BEYOND ENERGY: WHEN SUSTAINABILITY DIMENSIONS RESHAPE LLM SERVING DECISIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) serving has environmental impacts across energy consumption, carbon emission, water consumption, and biodiversity loss. Yet these dimensions are largely evaluated in isolation, leaving it unclear when and how they lead to different optimization decisions. We present PRISM, a unified framework for characterizing and optimizing LLM serving across energy, carbon, water, and biodiversity impacts. Our analysis reveals a fundamental distinction: computing configurations determine energy consumption, whereas where and when LLM serving is deployed determine its carbon, water, and biodiversity impacts. Under a fixed deployment choice and operational-only accounting, all dimensions preserve the same energy-based configuration ranking. Deployment rankings can diverge across dimensions, while embodied impacts can break configuration invariance when they exceed a lifecycle crossover boundary. PRISM identifies these conditions, quantifies cross-dimensional regrets, and balances the four dimensions. In regional-routing experiments, PRISM reduces median worst-case regret by 50.2% relative to the strongest baseline.

1 INTRODUCTION

The rapid adoption of large language models (LLMs) has raised growing concerns about their environmental impact (Lambert & Luccioni, 2026; Chien et al., 2026; Ding & Shi, 2024). These concerns initially centered on the substantial **energy consumption** (Strubell et al., 2019; Fernandez et al., 2025; Patel et al., 2024; Stojkovic et al., 2025) of LLM serving and later expanded to the associated **carbon emissions** (Patterson et al., 2021; Luccioni et al., 2023). Recent work further highlighted the substantial **water consumption** (Li et al., 2024b; Wu et al., 2025a) from LLM systems through datacenter cooling, electricity generation, and hardware manufacturing. Their lifecycle activities, such as resource extraction and land use, can also contribute to **biodiversity loss** (Shi et al., 2025a; Shi & Ding, 2026). Together, energy, carbon, water, and biodiversity capture distinct aspects of LLM serving sustainability (Shi et al., 2026).

Most existing sustainable AI research nevertheless evaluates LLM systems through a single environmental dimension. However, the four dimensions depend on different factors. Energy depends on workloads, models, hardware, and serving configurations (Chung et al., 2026; Shi & Ding, 2025); carbon additionally depends on the electricity mix and embodied emissions (Nguyen et al., 2024; Li et al., 2024c); water depends on cooling types, electricity-generation technologies, manufacturing, and local water scarcity (Wu et al., 2025a; Jiang et al., 2025a); and biodiversity aggregates multiple pathway-specific effects on ecosystems (Shi et al., 2025a; Shi & Ding, 2026). Although interconnected, these dimensions are neither equivalent nor necessarily proportional.

Beyond studying these dimensions separately, existing work provides limited insight into how they affect decisions. Prior sustainability-aware cloud systems demonstrate carbon–water tradeoffs (Jegham et al., 2025; Jiang et al., 2025b) and the benefits of regional workload shifting (Gsteiger et al., 2024), but typically consider only one or two dimensions, target general cloud workloads, and do not explain when or why computing-configuration and deployment rankings agree or diverge for LLM serving workloads. We therefore ask two central questions: ❶ When do energy, carbon, water, and biodiversity agree on computing configurations and deployment choices, and what causes their rankings to diverge? ❷ When they disagree, how should an LLM serving system balance the four dimensions while satisfying performance and quality requirements?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We present PRISM, a unified framework for characterizing and optimizing LLM serving across energy, carbon, water, and biodiversity. PRISM evaluates the four dimensions using consistent system boundaries and functional units across workloads, computing configurations, and deployment choices. It analyzes configuration and deployment rankings, identifies lifecycle crossover conditions, quantifies the disagreement through cross-dimensional regret, and optimizes deployment choices under multi-dimensional objectives. This paper makes the following contributions:

- We develop PRISM, a unified framework for characterizing and optimizing energy, carbon, water, and biodiversity impacts of LLM serving under consistent system boundaries and functional units.
- We establish operational configuration invariance: under a fixed deployment choice and operational-only accounting, all four dimensions preserve the configuration ranking induced by IT energy. We further derive a lifecycle crossover condition that identifies when configuration-dependent embodied impacts break this invariance.
- We systematically characterize the four dimensions across workloads, computing configurations, deployment choices, and quantify when their rankings agree or diverge.
- We formulate multidimensional deployment optimization that balances the four dimensions by limiting the largest relative penalty without directly combining impacts.

Our results reveal a fundamental distinction between computing configurations and deployment choices. Under a fixed deployment choice, computing configurations change impact magnitude but preserve the same operational configuration ranking across dimensions; embodied impacts change the minimum-impact configuration in only 4 of 882 evaluated settings. In contrast, deployment rankings exhibit stronger disagreement: among six representative regions, selecting the energy-minimizing region incurs 194% higher carbon emissions, 4,637% higher water impact, and 195% higher biodiversity impact than minimizing each respective dimension. When balancing all four dimensions, PRISM achieves the lowest median worst-case regret among the evaluated methods, with a 50.2% median reduction relative to the strongest baseline. These findings show that energy can guide configuration selection under fixed deployment and operational-only accounting, but cannot serve as a reliable proxy for other impacts when selecting where and when to deploy LLM serving.

2 Multidimensional Impact Model and Decision Properties

We first unify established accounting methods for energy (E), carbon (C), water (W), and biodiversity (B) under a common formulation and then derive their implications for LLM serving. Detailed accounting methods for each dimension are provided in Appendix A.1.

Let w denote the *workload*, including the requests or tasks to be served and the functional unit (Wu et al., 2025b) used for comparison. We vary workloads to study how their characteristics affect environmental impacts. Let x denote a *computing configuration*, including the model, GPU type, GPU count, and parallelism. A *deployment choice* (r, t) specifies the region r and execution time period t. We distinguish between **operational**, **embodied**, and **lifecycle** impacts (Gupta et al., 2022). Operational impact arises from energy consumed while running the workload. Embodied impact arises from manufacturing the hardware and is allocated to the workload based on its use of that hardware. Within our system boundary, lifecycle impact refers to the sum of operational and embodied impacts. Our lifecycle system boundary (Suh et al., 2004) includes hardware manufacturing and system operation; transportation and end-of-life are excluded because detailed inventory data are unavailable. We first measure the IT energy consumed by workload w under configuration x, denoted by $E_{\mathrm{IT}}(x, w)$. The total operational energy attributed to the workload is

$$E_{\mathrm{op}}(x, r, t, w) = E_{\mathrm{IT}}(x, w) \cdot \mathrm{PUE}(r, t), \tag{1}$$

where $\mathrm{PUE}(r, t)$ denotes the Power Usage Effectiveness (PUE) of a datacenter in region r during time t, defined as the ratio of total facility energy consumption to IT equipment energy consumption (Barroso et al., 2019). Although the four sustainability dimensions characterize different outcomes, their operational components share a common structure:

$$I_{m,\mathrm{op}}(x, r, t, w) = E_{\mathrm{IT}}(x, w) \cdot \alpha_m(r, t), \qquad m \in \{E, C, W, B\}, \tag{2}$$

where $\alpha_m(r, t)$ is the *operational impact intensity* of dimension m, defined as the operational impact produced per unit of IT energy. It is derived from three types of regional and temporal data: facility factors, such as PUE and Water Usage Effectiveness (WUE) (Li et al., 2024b); electricity-system environmental intensities, such as carbon intensity (CI) (Maji et al., 2022; Yan et al., 2025), electricity

Table 1: Unified operational and embodied accounting across four sustainability dimensions.

Dimension	Effective Operational Intensity $\alpha_m(r,t)$	Embodied Component	Output
Energy	$\mathrm{PUE}(r,t)$	—	kWh
Carbon	$\mathrm{PUE}(r,t)\cdot\mathrm{CI}(r,t)$	$C_{\mathrm{emb}}(x,w)$	kg CO_2e
Water	$[\mathrm{WUE}(r,t)+\mathrm{PUE}(r,t)\cdot\mathrm{EWIF}(r,t)]\cdot\mathrm{WSF}(r)$	$W_{\mathrm{emb}}(x,w)$	m^3 world-eq
Biodiversity	$\mathrm{PUE}(r,t)\cdot\mathrm{BIF}(r,t)+\mathrm{WUE}(r,t)\cdot\mathrm{WSF}(r)\cdot\mathrm{CF}_{B,W}(r)$	$B_{\mathrm{emb}}(x,w)$	species·year

water intensity (EWIF) (Wu et al., 2025a), and grid biodiversity intensity (BIF) (Shi et al., 2025a); and local characterization factors. Specifically, water stress factor (WSF) (Wu et al., 2025a) adjusts water consumption for local water stress, while $\mathrm{CF}_{B,W}$ converts direct local water consumption into biodiversity impact (Shi & Ding, 2026). We refer to these inputs collectively as *environmental data*. The LLM serving workload and computing configuration determine $E_{\mathrm{IT}}(x,w)$, while deployment choice determines $\alpha_m(r,t)$. Adding the embodied component gives the lifecycle impact

$$I_m(x,r,t,w) = E_{\mathrm{IT}}(x,w)\cdot\alpha_m(r,t) + I_{m,\mathrm{emb}}(x,w). \tag{3}$$

Table 1 summarizes how the four dimensions instantiate this common model. The complete derivations of each intensity, factor, and embodied component are in Appendix A.1.

Within dimension m, decisions are ranked by increasing impact. A *configuration ranking* compares x while holding (w,r,t) fixed, whereas a *deployment ranking* compares (r,t) while holding (x,w) fixed. Two dimensions disagree when they reverse the ordering of at least one pair of configurations or deployment choices.

Proposition 1 (Operational Configuration Invariance). For a fixed deployment choice, operational sustainability dimensions preserve the configuration ranking induced by IT energy:

$$E_{\mathrm{IT}}(x_i,w) < E_{\mathrm{IT}}(x_j,w) \iff I_{m,\mathrm{op}}(x_i,r,t,w) < I_{m,\mathrm{op}}(x_j,r,t,w), \tag{4}$$

where x_i and x_j are two computing configurations. This follows because $\alpha_m(r,t)$ is positive and constant across computing configurations within the same deployment choice.

Proposition 2 (Operational Deployment Ranking Invariance). For a fixed workload and sustainability dimension, the operational deployment ranking is independent of LLM computing configurations. For two choices (r_i,t_i) and (r_j,t_j),

$$I_{m,\mathrm{op}}(x,r_i,t_i,w) < I_{m,\mathrm{op}}(x,r_j,t_j,w) \iff \alpha_m(r_i,t_i) < \alpha_m(r_j,t_j). \tag{5}$$

The common positive factor $E_{\mathrm{IT}}(x,w)$ cancels when comparing regions. Changing LLM computing configuration therefore changes the magnitude of operational impact but not the deployment ranking.

Proposition 3 (Lifecycle Crossover). Configuration-dependent embodied impacts can break operational configuration invariance. Consider configurations x_i and x_j such that $E_{\mathrm{IT}}(x_i,w) < E_{\mathrm{IT}}(x_j,w)$. Dimension m prefers the less energy-efficient configuration x_j when

$$I_{m,\mathrm{emb}}(x_i,w) - I_{m,\mathrm{emb}}(x_j,w) > \alpha_m(r,t)\left[E_{\mathrm{IT}}(x_j,w) - E_{\mathrm{IT}}(x_i,w)\right]. \tag{6}$$

This condition defines the crossover boundary at which the embodied-impact advantage of x_j exceeds its operational disadvantage compared to x_i.

3 PRISM: Unified Characterization and Decision Optimization

Figure 1 presents PRISM, our framework for characterizing LLM serving and optimizing its deployment across energy, carbon, water, and biodiversity. PRISM connects four stages: input specification, unified characterization, cross-dimensional analysis, and deployment optimization. Additional details on PRISM's implementation and decision analysis are provided in Appendix A.2.

Input Specification and Unified Characterization. PRISM takes as input a workload context w, design space, deployment choices, service requirements, and the regional and lifecycle data required for environmental characterization. PRISM profiles each computing configuration to measure IT energy, latency, throughput, and output quality or task success, retaining only configurations that satisfy the same service requirements. It then combines the measured system profile with environmental data to calculate energy, carbon, water, and biodiversity impacts using models in §2. All dimensions use the same workload execution, system boundary, and functional unit, with impacts reported per request or per successfully completed task. Let $d=(x,r,t)$ denote a serving decision. For $m\in\{E,C,W,B\}$, $I_m(d,w)\equiv I_m(x,r,t,w)$ denotes the impact of executing w under d.

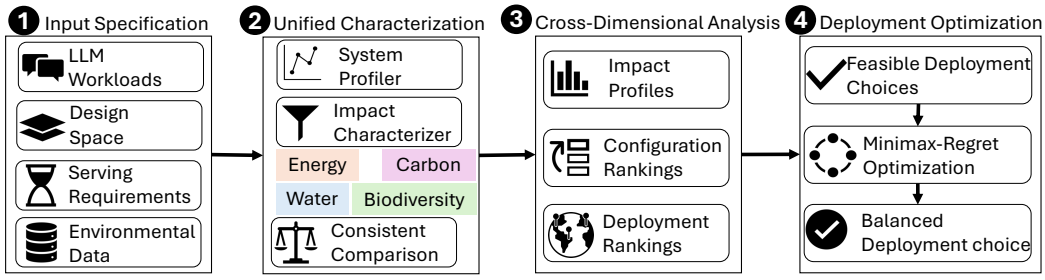


Figure 1: Overview of PRISM.

Cross-Dimensional Analysis. PRISM analyzes the four sustainability dimensions along two decision axes: computing configuration and deployment choice. For a fixed deployment choice, it compares configuration rankings across four dimensions and evaluates the operational configuration-invariance property in Proposition 1. For lifecycle analysis, it applies Proposition 3 to calculate the embodied-impact difference required to reverse each observed operational ranking. For a fixed computing configuration, it compares deployment rankings to determine when the four dimensions lead to different deployment choices. To quantify deployment disagreement, let d_i^* denote the deployment minimizing dimension i. The regret incurred under dimension j when selecting d_i^* is

$$R_{i\to j}(w) = \frac{I_j(d_i^*, w) - I_j(d_j^*, w)}{I_j(d_j^*, w)}. \tag{7}$$

A small $R_{i\to j}$ indicates that optimizing dimension i produces a deployment choice close to the optimum for dimension j.

Deployment Optimization. According to Proposition 1, all four dimensions preserve the energy-based configuration ranking under the same deployment choice. PRISM first selects the minimum-energy feasible configuration x_E^* for each workload. Let $s \in \mathcal{S}$ denote a deployment choice, where $\mathcal{S}$ contains all feasible deployment choices subject to service and capacity constraints. For a single workload, s reduces to one serving decision $d = (x_E^*, r, t)$; for a workload trace, it assigns each request k to a region and execution time period. The aggregate impact of a deployment choice is

$$I_m(s) = \sum_k I_m(x_E^*(w_k), r_k, t_k, w_k), \qquad I_m^* = \min_{s\in\mathcal{S}} I_m(s). \tag{8}$$

For multi-dimensional optimization, PRISM identifies the Pareto-efficient deployment and selects

$$s^*_{\text{PRISM}} = \arg\min_{s\in\mathcal{S}} \max_{m\in\{E,C,W,B\}} \frac{I_m(s) - I_m^*}{I_m^*}. \tag{9}$$

This formulation minimizes the largest relative loss from the dimension-specific optimum, allowing the four dimensions to be balanced without aggregating into one single objective.

4 Evaluation Methodology

We evaluate PRISM through three research questions (RQs). **RQ1 (Characterization):** How do LLM workloads, configurations, and deployment choices affect the magnitude and variation of energy, carbon, water, and biodiversity impacts? **RQ2 (Ranking Disagreement)**: When do sustainability dimensions lead to different configuration or deployment rankings, and what causes these difference? **RQ3 (Multi-dimensional Optimization):** How effectively does PRISM balance the four dimensions while satisfying performance and quality requirements?

Setup. We evaluate both non-agentic and agentic LLM serving workloads. ShareGPT (ShareGPT, 2022) represents interactive conversations, LongBench (Bai et al., 2023) represents long-context tasks, RepoBench (Liu et al., 2024) represents IDE-level code completion tasks, and SWE-bench Verified (Jimenez et al., 2024) represents agentic software-engineering tasks. Our design space includes models from the Llama-3 (Grattafiori et al., 2024), GPT-OSS (OpenAI, 2025), Qwen3 (Qwen Team, 2025), and Gemma-4 (Google DeepMind, 2026) families, deployed on NVIDIA A100, L40, and H100 GPUs. We vary the model, GPU, parallelism, and request load while retaining only configurations that satisfy performance and quality requirements. GPU energy is measured using

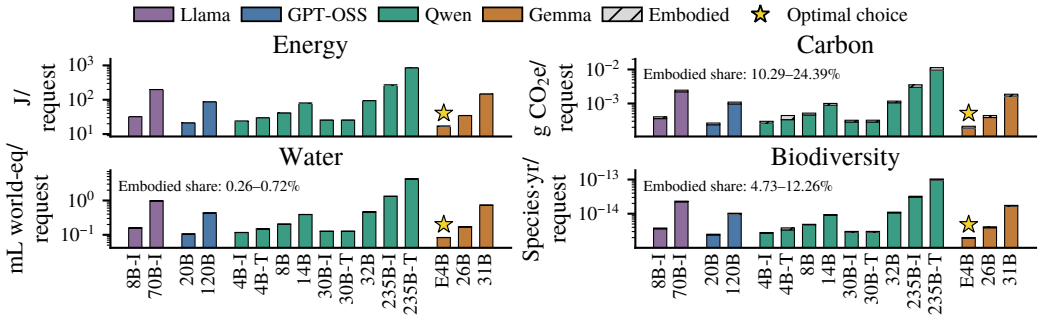


Figure 2: Impact of model family and size on per-request energy, carbon, water, and biodiversity. Solid and hatched bars denote operational and embodied impacts, respectively; stars mark the minimum-impact configuration for each dimension.

NVML (NVIDIA Corporation, 2025), and non-GPU host energy is estimated from component utilization. We report impacts per request for non-agentic workloads and per successfully completed task for agentic workloads. We evaluate deployment across 46 locations (Table 9) worldwide based on major cloud providers' official region documentation to span contrasting facility efficiency, electricity mixes, carbon and water intensities, water stress, and ecological conditions.

Key Assumptions. We compare computing configurations and deployment choices that deliver workload outcomes under the same functional unit and service requirements. We assume that every evaluated computing configuration is available in all deployment choices and that a fixed computing configuration has the same IT-level execution profile across regions; regional conditions affect its impact through operational impact intensities and environmental data. Our optimization focuses on operational impacts because embodied impacts are already incurred for an existing hardware fleet and cannot be changed through workload scheduling or regional placement; including them in the optimization can therefore increase, rather than reduce, total environmental impact (Bashir et al., 2024; Gsteiger et al., 2024). Complete workload statistics, model and hardware specifications, profiling and quality-evaluation protocols, quality scores, latency constraints, and regional data are provided in Appendix A.3.

5 Characterizing Four Dimensions

We first examine how workloads, computing configurations, and deployment choice affect energy, carbon, water, and biodiversity impacts. Beyond comparing their impact magnitudes, we ask whether the four dimensions rank the same computing configurations consistently.

Configurations and Workloads. Figure 2 provides a detailed characterization across model families and scales. We fix the workload to ShareGPT, use H100 GPUs, select the most energy-efficient tensor parallelism (TP) for each model, and apply France's 2024 annual-average environmental intensities. Impact generally increases with model size because larger models require more serving energy per request. Model size alone, however, does not determine impact: mixture-of-experts models can incur substantially lower impact than similarly sized dense models because they activate only a subset of their parameters per token. Despite measuring different sustainability outcomes, the four dimensions exhibit nearly identical trends and select the same minimum-impact model in this setting. Operational impact dominates carbon, water, and biodiversity, and their values therefore largely scale with the same underlying serving energy. The allocated embodied components change their magnitudes but are insufficient to reverse the preferred configuration in this example.

Figure 3 summarizes whether this pattern generalizes across model and scale, GPU platform, TP, and workload types. These factors substantially change impact magnitude. H100 generally achieves lower per-request impact than A100 and L40, whereas the effect of TP depends on scaling efficiency: additional GPUs reduce impact only when their throughput improvement offsets the added energy consumption. Workload properties also matter. The same model and hardware configuration show a large variance of per-request impact among workloads with distinct prompt and generation length distributions (Table 4), resulting in up to 23.8× ratio between the shortest chat conversations and the longest document summarizations. Agentic workloads are excluded from this comparison because

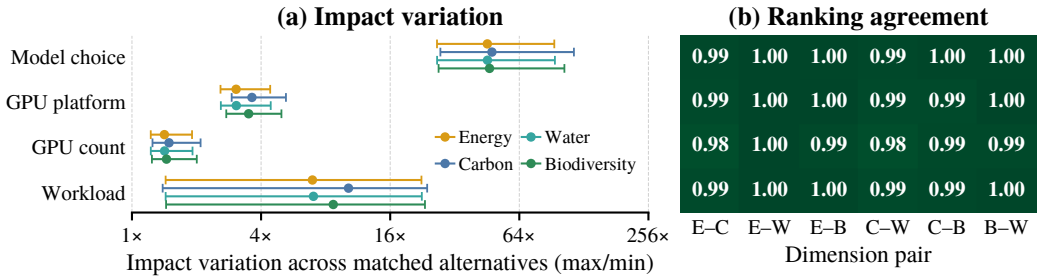


Figure 3: Summary of computing configuration and workload effects under a fixed deployment choice. (a) Median and interquartile range of impact variation when varying model choice, GPU platform, GPU count, or non-agentic workload. (b) Cross-dimensional rank agreement for the same comparisons. Each comparison varies only the indicated category and uses a same functional unit.

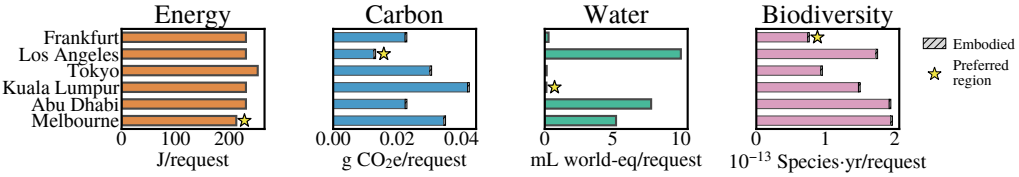


Figure 4: Regional variation in per-request sustainability impacts for Qwen3-235B-A22B-Instruct on H100 (TP8). Stars mark the lowest-impact region for each dimension among the locations shown.

their impacts are measured using a different functional unit: impact per successfully completed task. Appendix A.4.2 shows that SWE-Verified incurs much larger per-task impact due to repeated model invocations and long execution, while lower task success can cause a smaller model to have higher impact per successful task by amortizing failed attempts over fewer successes.

Nevertheless, under fixed deployment choices, carbon, water, and biodiversity impacts remain strongly aligned with the energy-based ranking across computing configurations. Computing configurations and workloads determine how much impact is produced, but the deployment choice among sustainability dimensions usually does not change which configuration is preferred. Detailed results for models, individual GPUs, TP levels, and workloads are provided in Appendix A.4.1 and A.4.2.

Takeaway 1: *Computing configurations substantially change the magnitudes of impacts in all four dimensions, but under fixed deployment choices, operational impacts of carbon, water, and biodiversity generally preserve the energy-based configuration ranking.*

Region and Time. Figure 4 fixes the workload and computing configuration while varying the deployment region. We show six representative locations from our 46-location dataset using 2024 annual-average operational impact intensities. Energy changes the modestly because the IT-level execution is fixed and regional variation enters primarily through PUE. Carbon, water, and biodiversity impacts vary much more because they additionally depend on the electricity mix, cooling conditions, water stress, and operational impact intensities. These regional factors also produce different preferences. Among the locations shown, Melbourne minimizes energy, Los Angeles carbon, Kuala Lumpur water, and Frankfurt biodiversity. Thus, a region with favorable facility efficiency does not necessarily minimize its broader environmental impacts. The computing configuration determines the underlying IT energy demand, whereas the deployment choice determines how that demand translates into carbon, water, and biodiversity impacts. Regional preferences are also time-dependent. Monthly and hourly measurements (Figure 20) exhibit ranking crossovers, indicating that a region preferred under annual-average conditions may not remain preferable at finer temporal resolutions. Detailed temporal results and the corresponding operational impact intensity trends are provided in Appendix A.4.3.

Takeaway 2: *Unlike configuration rankings, which are aligned across dimensions under operational accounting, deployment rankings can differ across dimensions, causing energy, carbon, water, and biodiversity to prefer different regions or execution times.*

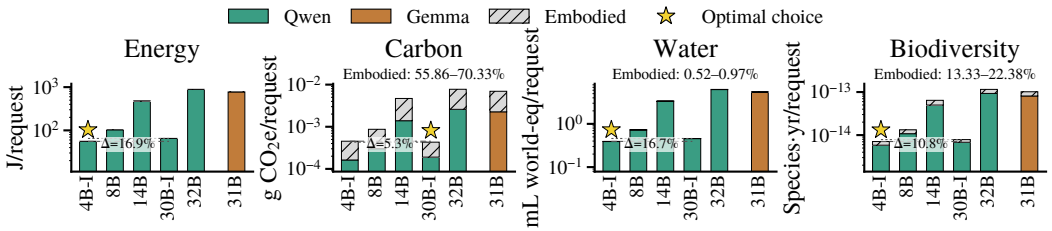


Figure 5: Lifecycle-induced configuration ranking disagreement for RepoBench in Norway with edit-similarity ≥ 44.65. Stars mark the minimum-impact configuration for each dimension.

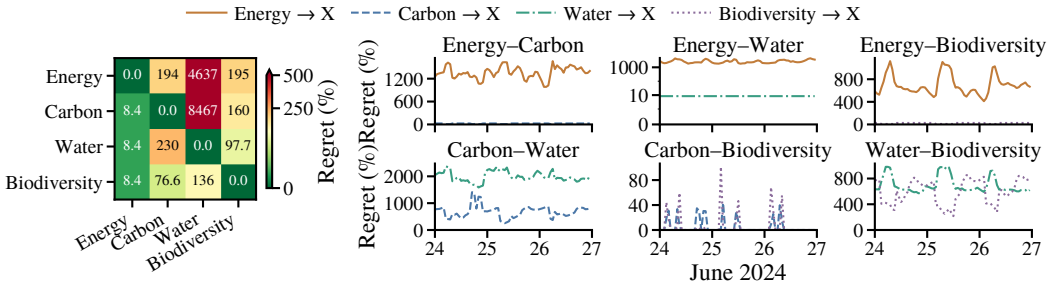


Figure 6: Left: directional cross-dimensional regret among the dimension-optimal regions in Figure 4. Cell (i, j) reports the additional impact under dimension j when selecting the region optimized for dimension i. Right: hourly directional cross-regret across selected regions during June 24–26, 2024, using the spatiotemporal operational impact intensities in Figure 20.

6 When Sustainability Dimensions Disagree

The preceding results show that, under a fixed deployment choice, operational impacts in all four dimensions preserve the configuration ranking induced by IT energy, but their deployment rankings can differ. We next examine two sources of cross-dimensional ranking disagreement. First, configuration-dependent embodied impacts can break operational configuration invariance when the lifecycle crossover condition in Proposition 3 is satisfied. Second, regional and temporal variation in operational impact intensities can cause the dimensions to produce different deployment rankings.

Lifecycle-Induced Configuration Disagreement. Including configuration-dependent embodied impacts can break operational configuration invariance. Figure 5 shows the largest disagreement among minimum-impact configurations observed in our quality-constrained evaluation. Energy, water, and biodiversity rank Qwen3-4B-Instruct on one H100 with TP1 first, whereascarbon ranks Qwen3-30B-Instruct on two H100s with TP2 first. The latter consumes 16.9% more energy per request and increases water and biodiversity impacts by 16.7% and 10.8%, respectively, but produces lower carbon emissions. This carbon ranking reversal occurs because Norway's low carbon intensity makes the carbon impact from the additional energy consumption relatively small. Meanwhile, the higher throughput of Qwen3-30B amortizes its embodied carbon emission over more requests. For this configuration pair, the embodied-carbon advantage of Qwen3-30B is $1.83\times$ its additional operational-carbon penalty and therefore exceeds the lifecycle crossover boundary.

More generally, lifecycle configuration rankings can diverge only when the embodied-impact difference favors the higher-energy configuration and is large enough to exceed its region-specific operational penalty. The embodied share of total impact alone therefore does not predict a ranking reversal. Configuration ranking disagreement is rare in our evaluation: the dimensions select different minimum-impact configurations in only 4 of 882 quality- and performance-constrained settings (0.45%). Appendix A.5 reports the remaining cases, controlled comparisons of GPU and parallelism choices, complete crossover calculations, and sensitivity to the assumed hardware lifetime.

Takeaway 3: *Lifecycle configuration rankings diverge only when the embodied-impact advantage of a higher-energy configuration exceeds its region-specific operational penalty. A large embodied share alone does not imply a ranking reversal.*

Regional and Temporal Deployment Ranking Disagreement. The four dimensions can produce different deployment rankings even under operational-only accounting because their operational

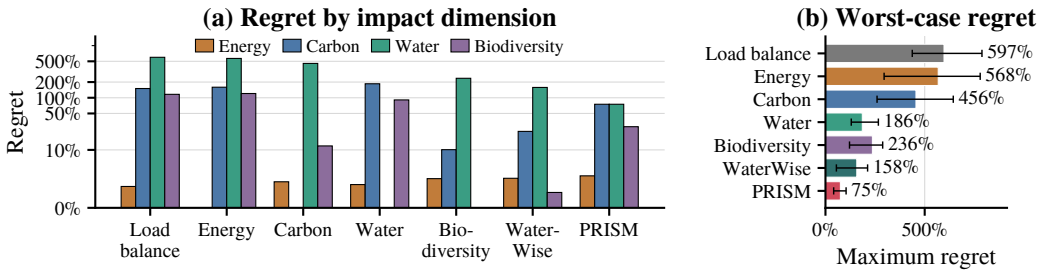


Figure 7: Environmental trade-offs in offline regional routing. Regret measures the relative increase over the best feasible value for each impact. (a) Median regret for each dimension across 20 regional-capacity seeds; the symlog axis is linear below 10%. (b) Median worst-case regret, with whiskers showing the full range across seeds. Lower is better.

impact intensities vary across regions. Figure 6 (left) quantifies this disagreement among the six representative regions using directional cross-dimensional regret. Relative to the minimum-impact region under each dimension, choosing the energy-minimizing region increases carbon emissions by 194%, water impact by 4,637%, and biodiversity impact by 195%. Minimizing operational energy is therefore not a reliable proxy for minimizing carbon, water, or biodiversity impacts when selecting a deployment region. These penalties are also strongly asymmetric. Although the energy-minimizing region performs substantially worse under the other three dimensions, the regions ranked first by carbon, water, and biodiversity increase operational energy by no more than 8.4%. Thus, a modest increase in energy can correspond to a much larger reduction in another environmental impact.

Execution time also affects the consequences of deployment ranking disagreement. Figure 6 (right) shows that directional regret changes considerably across hours as operational impact intensities vary. Some pairs, particularly energy and water, exhibit consistently high regret throughout the evaluated period, whereas the penalties between other pairs depend more strongly on execution time. Temporal scheduling can therefore reduce or increase the penalty of following one dimension's deployment ranking over another, but does not necessarily eliminate the underlying disagreement.

Takeaway 4: *Under operational-only accounting, cross-dimensional disagreement arises primarily in deployment rankings. Following the energy ranking can incur large and asymmetric carbon, water, and biodiversity penalties, while execution time changes the severity of these penalties but does not necessarily eliminate them.*

7 Can PRISM Balance Multiple Sustainability Dimensions?

We evaluate how effectively PRISM balances all four dimensions. We first instantiate Equation (9) as an offline regional-routing problem in which workload demand and hourly environmental data are known in advance. Its solution provides an oracle benchmark for evaluating rolling-horizon (Sethi & Sorger, 1991) online routing with limited future information. We use the first 24 hours of the Azure LLM Inference Dataset 2024 (Stojkovic et al., 2025), combining code and conversation requests with equal offered GPU demand. Each request is assigned within its arrival hour to one of the twelve regions in Figure 20, subject to regional capacity constraints. We use hourly operational impact intensities and generate heterogeneous regional capacities across 20 random seeds. We compare environment-unaware geographical load balancing; routing that individually minimizes energy, carbon, water, or biodiversity; an offline adaptation of WaterWise (Jiang et al., 2025b) that jointly optimizes carbon and water; and PRISM that minimizes the largest normalized regret across four dimensions. We further evaluate rolling-horizon PRISM and extend it to jointly select computing configurations and deployment regions for an agentic workload with completion time included as an additional objective. Trace construction, capacity generation, optimization formulations, baseline implementations, solver runtime, and sensitivity analyses are described in §A.6.

Balancing Four Dimensions. Figure 7 shows that minimizing one dimension can impose a large penalty on another. Median worst-case regret ranges from 185.5% for water-only routing to 568.0% for energy-only routing, while environment-unaware load balancing reaches 597.2%. WaterWise lowers median worst-case regret to 157.8%, but its largest remaining penalty is its water regret. PRISM achieves the lowest median worst-case regret of 75.1%, with a 50.2% median reduction relative to WaterWise. This improvement does not mean that PRISM minimizes every dimension

individually. Compared with WaterWise, PRISM accepts higher carbon regret (75.1% instead of 22.7%) but lowers water regret from 157.8% to 75.1%. It therefore selects a more balanced deployment by preventing any one dimension from incurring a disproportionately large penalty.

Online Routing and Joint Optimization. Sections A.7.1 and A.7.2 evaluates PRISM beyond offline regional routing. With a one-hour planning horizon, rolling-horizon PRISM achieves an objective value within 0.7% of the offline oracle and remains stable when the forecast operational impact intensities contain 20% error. We also jointly optimize computing configuration and deployment region for an agentic workload. Including task-completion time as a fifth objective changes the selected configuration mix and limits the maximum regret across the four dimensions and completion time to 77.3%. This result shows that performance requirements can change the preferred computing and deployment choices and should therefore be considered jointly with environmental objectives.

Takeaway 5: *No policy minimizes all dimensions simultaneously.* PRISM *does not make every dimension optimal; instead, it prevents any dimension from becoming disproportionately poor.*

8 Related Work

Environmental Impacts of LLMs. Prior work has extensively studied the energy consumption (Strubell et al., 2019; Fernandez et al., 2025; Patel et al., 2024) and carbon emissions of LLM training and serving (Patterson et al., 2021; Luccioni et al., 2023; Ding & Shi, 2024; Lambert & Luccioni, 2026; Chien et al., 2026). Recent LLM studies further characterize how workload properties, model scale, hardware platforms, parallelism, and serving load affect inference energy and carbon (Li et al., 2023; Nguyen et al., 2024; Stojkovic et al., 2025; Shi et al., 2025b; Chung et al., 2026; Shi & Ding, 2025). Beyond carbon, emerging work quantifies water consumption from datacenter cooling, electricity generation, and hardware manufacturing (Li et al., 2024b; Wu et al., 2025a; Jiang et al., 2025a), while lifecycle-based studies incorporate embodied hardware impacts (Gupta et al., 2022; Li et al., 2024c;a). Biodiversity impact captures ecosystem damage from computing-related emissions, water consumption, land use, and other lifecycle pathways (Shi & Ding, 2026; Shi et al., 2025a). A few studies report multiple environmental dimensions for the same AI or LLM systems (Jegham et al., 2025; Jiang et al., 2025b), but primarily compare impact magnitudes rather than analyzing their decision implications. Our work instead examines when the four dimensions preserve the same decisions, when their rankings diverge, and what causes that divergence.

Sustainability-Aware Serving and Scheduling. Sustainability-aware systems shift workloads across locations or times based on electricity availability and carbon intensity (Radovanović et al., 2022; Acun et al., 2023; Hanafy et al., 2024; Tian et al., 2026). Caribou uses geospatial shifting to reduce the operational carbon emissions of serverless applications (Gsteiger et al., 2024), while WaterWise jointly optimizes carbon and water for geographically distributed workloads (Jiang et al., 2025b). These systems demonstrate the benefits of regional placement and potential conflicts between environmental objectives. However, they generally consider only one or two dimensions, target general cloud workloads, and do not explain when or why computing configuration and deployment rankings agree or diverge for LLM serving. PRISM instead analyzes all four dimensions under a common impact model and balances them without directly combining their different values.

9 Conclusion and Limitations

We presented PRISM, a unified framework for characterizing and optimizing LLM serving across energy, carbon, water, and biodiversity. Our results show that computing configurations primarily determine impact magnitude, whereas deployment choices and embodied impacts can cause the dimensions to prefer different decisions. We hope this work encourages sustainable LLM serving to consider environmental dimensions beyond energy when making deployment decisions.

Limitations. We assume that the same computing configurations are available across regions and that a fixed configuration has the same IT-level execution profile in every region. Because regional capacities are not publicly available, routing experiments use synthesized heterogeneous capacities. Lifecycle results are sensitive to assumptions about hardware lifetime, utilization, and embodied-impact allocation, while deployment results inherit uncertainty from available environmental data.

AI USE STATEMENT

We used generative AI tools, including ChatGPT and Codex, to help refine the conceptual framework, mathematical claims and their analytical justifications, research hypotheses, experimental methodology, method implementation, dataset preparation, and interpretation of results. We have not used generative AI tools for assisting with translation. Generating synthetic datasets and conducting qualitative or thematic data analysis are not applicable to this work.

Additionally, generative AI was used to help implement and review research code; create and modify scientific figures; suggest experimental parameters; draft and edit portions of the paper; improve readability; source public information used in environmental-intensity dataset preparation; identify supporting references for software, data sources, and modeled devices. Generative AI was not used to summarize or analyze prior literature as part of the scientific argument of this work.

All AI-assisted code, data processing, mathematical derivations, experimental results, figures, and manuscript text were reviewed by the authors. Numerical results were checked against the underlying measurements and analysis outputs, and externally sourced data and citations were verified against their original sources. The authors take responsibility for the final content of this work, including text, claims, code, data, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main text and appendix describe PRISM's sustainability accounting, optimization formulations, evaluation workloads, hardware configurations, quality criteria, and environmental factors. We provide source code, experiment interfaces, tests, and figure-generation scripts at https://anonymous.4open.science/r/PRISM-2653/. The repository documents installation, external dependencies, input schemas, and experiment execution, and includes selected result tables and commands for regenerating the corresponding figures. Appendix A.3.7 summarizes the environmental and lifecycle data sources used in our analysis and specifies their availability and redistribution constraints. Raw LLM-serving measurements and certain externally sourced environmental inputs are not redistributed with this submission; in particular, some environmental data are subject to provider licensing restrictions. The repository identifies these external prerequisites and clearly distinguishes results that can be regenerated from included artifacts from those that require re-running experiments with separately obtained measurements or environmental datasets.

REFERENCES

Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. Carbon explorer: A holistic framework for designing carbon aware datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Volume 2*, 2023.

Amazon Web Services. AWS Cloud – amazon sustainability. https://sustainability.aboutamazon.com/products-services/aws-cloud, a. Accessed: 2026-09-14.

Amazon Web Services. AWS Regions. https://docs.aws.amazon.com/global-infrastructure/latest/regions/aws-regions.html, b. Accessed: 2026-09-16.

AMD. AMD EPYC 7443 Processor Specifications. https://www.amd.com/en/products/processors/server/epyc/7003-series/amd-epyc-7443.html, a. Accessed: 2026-09-21.

AMD. AMD EPYC 7763 Processor Specifications. https://www.amd.com/en/products/processors/server/epyc/7003-series/amd-epyc-7763.html, b. Accessed: 2026-09-21.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2023.

Luiz André Barroso, Urs Hölzle, Parthasarathy Ranganathan, and Margaret Martonosi. *The datacenter as a computer: Designing warehouse-scale machines*. Springer, 2019.

Noman Bashir, Varun Gohil, Anagha Belavadi Subramanya, Mohammad Shahrad, David Irwin, Elsa Olivetti, and Christina Delimitrou. The sunk carbon fallacy: Rethinking carbon footprint metrics for effective carbon-aware scheduling. In *Proceedings of the 2024 ACM Symposium on Cloud Computing*, pp. 542–551, 2024.

Andrew A Chien, Udit Gupta, Shaolei Ren, Akshitha Sriraman, and Bill Tomlinson. Strategies and design for increasing ai sustainability. *Nature Reviews Clean Technology*, pp. 1–15, 2026.

Jae-Won Chung, Jeff J Ma, Ruofan Wu, Jiachen Liu, Oh Jun Kweon, Yuxuan Xia, Zhiyu Wu, and Mosharaf Chowdhury. The ml. energy benchmark: Toward automated inference energy measurement and optimization. *Advances in Neural Information Processing Systems*, 38, 2026.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michal Stechly, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. URL https://doi.org/10.5281/zenodo.11171501.

DeepSeek-AI. DeepSeek-V4.1-Flash: Smarter, Faster, More Efficient. https://www.deepseek.com/en/news/deepseek-v4-1-flash/, September 2026. Accessed: 2026-09-21.

Yi Ding and Tianyao Shi. Sustainable llm serving: Environmental implications, challenges, and opportunities. In *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, pp. 37–38. IEEE, 2024.

Electricity Maps. Electricity maps. Electricity Maps, 2024. URL https://app.electricitymaps.com/. Accessed: 2025-10-07.

European Commission Joint Research Centre (JRC) and Netherlands Environmental Assessment Agency. EDGAR_2024_GHG: Emissions database for global atmospheric research. https://edgar.jrc.ec.europa.eu/dataset_ghg2024, 2024. Accessed: 2025-05-19.

Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. Energy considerations of large language model inference and efficiency optimizations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32556–32569, 2025.

Google. Power usage effectiveness – google data centers. https://www.datacenters.google/efficiency/. Accessed: 2026-09-14.

Google Cloud. Regions and Zones. https://docs.cloud.google.com/compute/docs/regions-zones, a. Accessed: 2026-09-16.

Google Cloud. General-Purpose Machine Family for Compute Engine. https://cloud.google.com/compute/docs/general-purpose-machines, b. Accessed: 2026-09-21.

Google DeepMind. Gemma 4 Model Card. https://ai.google.dev/gemma/docs/core/model_card_4, April 2026. Accessed: 2026-05-14.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Viktor Urban Gsteiger, Pin Hong Long, Yiran Sun, Parshan Javanrood, and Mohammad Shahrad. Caribou: Fine-grained geospatial shifting of serverless applications for sustainability. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles*, pp. 403–420, 2024.

Pranjol Sen Gupta, Md Rajib Hossen, Pengfei Li, Shaolei Ren, and Mohammad A Islam. A dataset for research on water sustainability. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, pp. 442–446, 2024.

Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. ACT: Designing sustainable computer systems with an architectural carbon modeling tool. In *ISCA*, 2022. URL https://doi.org/10.1145/3470496.3527408.

Walid A Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. Going green for less green: Optimizing the cost of reducing cloud carbon emissions. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 479–496, 2024.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, 2026. URL https://doi.org/10.5281/zenodo.20953922.

Qi Huangfu and J. A. Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018. doi: 10.1007/s12532-017-0130-5.

Mark A.J. Huijbregts, Zoran J.N. Steinmann, Pieter M.F. Elshout, Geert Stam, Francesca Verones, Marisa Vieira, Anne Hollander, Michiel Zijp, and Rosalie van Zelm. ReCiPe 2016: A Harmonized Life Cycle Impact Assessment Method at Midpoint and Endpoint Level. Report I: Characterization. Technical report, RIVM National Institute for Public Health and the Environment, Bilthoven, The Netherlands, 2016.

Intel. Intel Ethernet Controller X550 Datasheet. https://cdrdv2-public.intel.com/333369/333369_X550_Datasheet_Rev2.7.pdf, a. Rev. 2.7, July 13, 2023. Accessed: 2026-09-21.

Intel. Intel Xeon Platinum 8480+ Processor Specifications. https://www.intel.com/content/www/us/en/products/sku/231746/intel-xeon-platinum-8480-processor-105m-cache-2-00-ghz/specifications.html, b. Accessed: 2026-09-21.

Intel. Intel Xeon Processor E5-2699 v4 Specifications. https://www.intel.com/content/www/us/en/products/sku/91317/intel-xeon-processor-e52699-v4-55m-cache-2-20-ghz/specifications.html, c. Accessed: 2026-09-21.

Nidhal Jegham, Marwan Abdelatti, Chan Young Koh, Lassad Elmoubarki, and Abdeltawab Hendawi. How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference. *arXiv preprint arXiv:2505.09598*, 2025.

Yankai Jiang, Raghavendra Kanakagiri, Rohan Basu Roy, and Devesh Tiwari. Thirstyflops: Water footprint modeling and analysis toward sustainable hpc systems. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2025a.

Yankai Jiang, Rohan Basu Roy, Raghavendra Kanakagiri, and Devesh Tiwari. Waterwise: Co-optimizing carbon-and water-footprint toward environmentally sustainable cloud computing. In *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, pp. 297–311, 2025b.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Katherine Lambert and Sasha Luccioni. From cradle to cloud: A life cycle review of ai's environmental footprint. In *The 2026 ACM Conference on Fairness, Accountability, and Transparency*, pp. 518–540, 2026.

Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. Clover: Toward sustainable ai with carbon-aware machine learning inference service. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, 2023.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Sprout: Green generative ai with carbon-efficient llm inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21799–21813, 2024a.

Pengfei Li, Jianyi Yang, Mohammad A. Islam, and Shaolei Ren. Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models. *Communications of the ACM*, 2024b.

Yueying Lisa Li, Omer Graif, and Udit Gupta. Towards carbon-efficient llm life cycle. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems*, 2024c.

Banruo Liu, Haoran Qiu, Íñigo Goiri, Rodrigo Fonseca, Ricardo Bianchini, and Esha Choukse. Agentic coding in the wild: Characterizing github copilot traces at production scale. *arXiv preprint arXiv:2608.00101*, 2026.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems, 2024. URL https://arxiv.org/abs/2306.03091.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of machine learning research*, 24 (253):1–15, 2023.

Diptyaroop Maji, Prashant Shenoy, and Ramesh K Sitaraman. Carboncast: multi-day forecasting of grid carbon intensity. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 198–207, 2022.

Microsoft. List of Azure Regions. https://learn.microsoft.com/en-us/azure/reliability/regions-list. Accessed: 2026-09-16.

Microsoft Azure. Measuring energy and water efficiency for microsoft datacenters. https://datacenters.microsoft.com/sustainability/efficiency/. Accessed: 2026-09-14.

Sophia Nguyen, Beihao Zhou, Yi Ding, and Sihang Liu. Towards sustainable large language model serving. In *ACM SIGENERGY Energy Informatics Review (EIR)*, 2024.

NVIDIA. NVIDIA ConnectX-6 InfiniBand/Ethernet Adapter Cards: Specifications. https://networking-docs.nvidia.com/connectx6vpihw/specifications, a. Accessed: 2026-09-21.

NVIDIA. NVIDIA ConnectX-7 Adapter Cards: Specifications. https://networking-docs.nvidia.com/connectx7hw/specifications, b. Accessed: 2026-09-21.

NVIDIA. NVIDIA A100 Tensor Core GPU Datasheet. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf, 2020. Accessed: 2026-05-10.

NVIDIA. NVIDIA H100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/h100/, 2022a. Accessed: 2026-05-10.

NVIDIA. NVIDIA L40 GPU for Data Center. https://www.nvidia.com/en-us/data-center/l40/, 2022b. Accessed: 2026-05-10.

NVIDIA. NVIDIA DGX SuperPOD: Next generation scalable infrastructure for ai leadership—reference architecture featuring nvidia dgx h100 systems. https://docs.nvidia.com/https:/docs.nvidia.com/dgx-superpod-reference-architecture-dgx-h100.pdf, September 2023. Document RA-11333-001 V11.

NVIDIA Corporation. Nvidia management library (nvml), 2025. URL https://developer.nvidia.com/management-library-nvml.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 207–222, 2024.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, MariEllen Cottman, and Walfredo Cirne. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems*, 38(2):1270–1280, 2022.

Paul Reig, Tianyi Luo, Eric Christensen, and Julie Sinistore. Guidance for calculating water use embedded in purchased electricity. *World Resources Institute*, 2020.

Jon Saad-Falcon, Avanika Narayan, Hakki Orhun Akengin, J Griffin, Herumb Shandilya, Adrian Gamarra Lafuente, Medhya Goel, Rebecca Joseph, Shlok Natarajan, Etash Kumar Guha, et al. Intelligence per watt: Measuring intelligence efficiency of local ai. *arXiv preprint arXiv:2511.07885*, 2025.

Georg Seitfudem, Markus Berger, Hannes Müller Schmied, and Anne-Marie Boulay. The updated and improved method for water scarcity impact assessment in lca, aware2. 0. *Journal of industrial ecology*, 29(3):891–907, 2025.

Suresh Sethi and Gerhard Sorger. A theory of rolling horizon decision making. *Annals of operations research*, 29(1):387–415, 1991.

ShareGPT. Sharegpt - share and save your conversations with ai. https://sharegpt.com/, 2022.

Tianyao Shi and Yi Ding. Systematic characterization of llm quantization: A performance, energy, and quality perspective. *arXiv preprint arXiv:2508.16712*, 2025.

Tianyao Shi and Yi Ding. Birds: Characterizing and understanding biodiversity impact of large language model serving. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026.

Tianyao Shi, Ritbik Kumar, Inez Hua, and Yi Ding. When servers meet species: A fab-to-grave lens on computing's biodiversity impact. *ACM SIGENERGY Energy Informatics Review*, 5(2):34–40, 2025a.

Tianyao Shi, Yanran Wu, Sihang Liu, and Yi Ding. Disaggregated speculative decoding for carbon-efficient llm serving. *IEEE Computer Architecture Letters*, 24(2):369–372, 2025b.

Tianyao Shi, Yanran Wu, Inez Hua, and Yi Ding. Sustainability of computing systems: A survey from environmental impact perspectives. 2026.

Cornel Soci, Hans Hersbach, Adrian Simmons, Paul Poli, Bill Bell, Paul Berrisford, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Raluca Radu, et al. The era5 global reanalysis from 1940 to 2022. *Quarterly Journal of the Royal Meteorological Society*, 150(764):4014–4048, 2024.

Jovan Stojkovic, Chaojie Zhang, Inigo Goiri, Josep Torrellas, and Esha Choukse. Dynamollm: Designing llm inference clusters for performance and energy efficiency. In *HPCA*, 2025.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3645–3650, 2019.

Sangwon Suh, Manfred Lenzen, Graham J. Treloar, Hiroki Hondo, Arpad Horvath, Gjalt Huppes, Olivier Jolliet, Udo Klüppel, Yoshiaki Kunugi, Reinhard Sager, Sangwon Suh, and Thomas Wiedmann. System boundary selection in life-cycle inventories using hybrid approaches. *Environmental science & technology*, 38(3):657–664, 2004.

Yuyang Tian, Desen Sun, Yi Ding, and Sihang Liu. Cache your prompt when it's green—carbon-aware caching for large language model serving. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 10(1):1–28, 2026.

Roberto Turconi, Alessio Boldrin, and Thomas Astrup. Life cycle assessment (lca) of electricity generation technologies: Overview, comparability and limitations. *Renewable and sustainable energy reviews*, 28:555–565, 2013.

United States Environmental Protection Agency (EPA). Emissions & generation resource integrated database (egrid), egrid2023rev1. https://www.epa.gov/egrid, 01 2025. Accessed: 2025-05-19.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Yanran Wu, Inez Hua, and Yi Ding. Not all water consumption is equal: A water stress weighted metric for sustainable computing. *ACM SIGENERGY Energy Informatics Review*, 5(2):84–90, 2025a.

Yanran Wu, Inez Hua, and Yi Ding. Unveiling environmental impacts of large language model serving: A functional unit view. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10560–10576, 2025b.

Leyi Yan, Linda Wang, Sihang Liu, and Yi Ding. Ensembleci: Ensemble learning for carbon intensity forecasting. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, pp. 208–212, 2025.

Patrick Zippenfenig. Open-meteo.com weather api, 2023. URL https://open-meteo.com/.

A Appendix

This appendix provides supporting definitions, methodological details, experimental settings, and additional results for the analysis in the main text. We first present the notation used throughout the paper, followed by the detailed lifecycle accounting of energy, carbon, water, and biodiversity (§A.1) and the implementation of PRISM's characterization and decision-analysis framework (§A.2). We then describe the workloads, models, testbeds, environmental data, and measurement methodology used in our experiments in §A.3. Finally, we provide supplementary characterization (§A.4) and cross-dimensional disagreement results (§A.5), additional details and sensitivity analyses for the multidimensional routing optimization (§A.6), and extensions to rolling-horizon routing and agentic workloads (§A.7).

Table 2: Notation used throughout the paper. Indices denote the corresponding workload, configuration, region, time, impact dimension, or request group unless stated otherwise.

Notation	Meaning	Notation	Meaning
E, C, W, B	Energy, carbon, water, and biodiversity dimensions.	$\alpha_m(r,t)$	Operational impact intensity of dimension m per unit IT energy.
m	Sustainability-dimension index, $m \in \{E, C, W, B\}$.	$\mathrm{PUE}(r,t)$	Power Usage Effectiveness.
w	Workload context and functional unit.	$\mathrm{CI}(r,t)$	Grid carbon intensity.
x	Computing configuration, including the model, GPU type, and parallelism.	$\mathrm{WUE}(r,t)$	Direct datacenter water use per unit IT energy.
r	Deployment-region index.	$\mathrm{EWIF}(r,t)$	Electricity water-intensity factor.
t	Execution-time or routing-interval index.	$\mathrm{WSF}(\ell)$	Water stress factor at location ℓ; $\mathrm{WSF}(r)$ is the datacenter-region value.
$d = (x, r, t)$	Complete serving decision.	$C_{\mathrm{emb}}, W_{\mathrm{emb}}, B_{\mathrm{emb}}$	Workload-allocated embodied carbon, water, and biodiversity impacts.
$\mathcal{X}$	Candidate configuration space.	I_W^{adj}	Water impact adjusted by local water stress.
$\mathcal{X}_{\mathrm{feas}}(w)$	Configurations satisfying service requirements for workload w.	W_ℓ, ℓ	Water consumed at location ℓ, and the location index.
L, T, Q	Latency, throughput, and output quality.	W_{dir}	Direct datacenter water consumption.
$L_{\max}, T_{\min}, Q_{\min}$	Latency, throughput, and quality service thresholds.	$\mathrm{BIF}(r,t)$	Biodiversity impact intensity of grid electricity.
$E_{\mathrm{IT}}(x,w)$	IT energy consumed by workload w under configuration x.	$p, q_p(r,t)$	Environmental-flow index and amount of flow p per unit grid electricity.
$E_{\mathrm{op}}(x,r,t,w)$	Facility-level operational electricity attributed to the workload.	$\mathrm{CF}_{B,p}$	Factor converting environmental flow p to ecosystem damage.
$I_m(x,r,t,w)$	Lifecycle impact in dimension m.	$\mathrm{CF}_{B,W}(r)$	Factor converting local water consumption to biodiversity damage.
$I_{m,\mathrm{elec}}, I_{m,\mathrm{dir}}, I_{m,\mathrm{emb}}$	Electricity-related, direct-resource, and embodied components of impact m.	ρ_m	Lifecycle crossover boundary multiple for dimension m.
$\mathbf{I}(d,w)$	Four-dimensional impact profile $[I_E, I_C, I_W, I_B]$.	ΔE_{IT}	IT-energy difference between the compared configurations.
x_i, x_j (or x_1, x_2)	Configurations compared in ranking/crossover analysis.	$\Delta I_{m,\mathrm{emb}}$	Embodied-impact difference between the compared configurations in dimension m.
d_i^*	Serving decision whose deployment choice minimizing dimension i fpr fixed x, w.	$x_E^*(w)$	Minimum-IT-energy feasible configuration for workload w.
$R_{i\to j}(w)$	Regret in dimension j from choosing the decision optimal for dimension i.	$R_m(s)$	Normalized regret of deployment plan s in dimension m.
q_{chat}	Relative chat-quality score from pairwise LLM judging.	$N_{\mathrm{win}}, N_{\mathrm{tie}}, N_{\mathrm{loss}}$	Pairwise judge win, tie, and loss counts.
$\mathcal{W} = \{w_k\}$	Workloads or request groups in a deployment instance.	w_k	Workload or request group k.
s	Feasible deployment plan.	$\mathcal{S}(\mathcal{W})$	Feasible deployment plans for $\mathcal{W}$; $\mathcal{S}$ is main-text shorthand.
g_k	Serving-capacity requirement of workload/request group k.	$K_{r,t}$	Available serving capacity in region r at time t.
$I_m(s,\mathcal{W})$	Aggregate impact of plan s in dimension m; $I_m(s)$ when $\mathcal{W}$ is implicit.	I_m^*	Best feasible value of impact dimension m.
$s^*_{\mathtt{PRISM}}$	Plan minimizing the maximum normalized environmental regret.	σ	Log-space standard deviation used to generate regional capacity shares.

Table 2: Notation used throughout the paper (continued).

Notation	Meaning	Notation	Meaning
y_{ir}	Fraction of request group i assigned to region r.	q_i, g_i, t_i	Request-equivalent count, GPU-seconds per request, and arrival hour of request group i.
z	Maximum-regret variable in the offline routing Linear Programming.	H	Rolling-horizon forecast length in hours.
$y_{ir}^{(x)}$	Fraction of request group i assigned to configuration x in region r.	$\mathcal{A}$	Set of agentic request groups.
τ_x	Mean completion time of successful agentic tasks under configuration x.	ω_i	Weight of agentic request group i in the completion-time objective.
$L(s), L^*, R_L(s)$	Mean completion time, its feasible optimum, and completion-time regret.	$I_m(s), I_m^*, R_m(s)$	Environmental impact, independent optimum, and regret for dimension m in the agentic extension.
$\Omega_{\mathcal{A}}$	Total agentic-group weight, $\sum_{i\in\mathcal{A}} \omega_i$.	$\mathcal{R}$	Candidate deployment-region set

Notation. Table 2 summarizes the notation used throughout the main text and appendix. It covers the workload, configuration, region, and time indices; lifecycle impact quantities and operational impact intensities; service constraints and cross-dimensional regret measures; and variables introduced in the deployment optimization and its extensions. Unless otherwise stated, the same notation and definitions are used consistently across the accounting, characterization, and decision-analysis formulations that follow.

A.1 Detailed Sustainability Accounting

The environmental impacts of LLM serving originate from three common sources: the electricity used to execute and support inference, direct resources consumed by the datacenter, and embodied impacts associated with computing hardware. We characterize these sources consistently across four sustainability dimensions: energy (E), carbon (C), water (W), and biodiversity (B).

Common Accounting Model. Let w denote an LLM serving workload executed using configuration x, in region r, during time interval t. Configuration x specifies the model, hardware, and serving parameters. We first measure the IT energy consumed by the workload, denoted by $E_{\mathrm{IT}}(x, w)$. The total operational electricity attributed to the workload is

$$E_{\mathrm{op}}(x, r, t, w) = E_{\mathrm{IT}}(x, w) \cdot \mathrm{PUE}(r, t), \tag{10}$$

where $\mathrm{PUE}(r, t)$ denotes the power usage effectiveness (PUE) of a datacenter in region r during time t, defined as the ratio of total facility energy consumption to IT equipment energy consumption.

For each sustainability dimension $m \in \{E, C, W, B\}$, we distinguish operational, direct, and embodied components:

$$I_m(x, r, t, w) = I_{m,\mathrm{elec}}(x, r, t, w) + I_{m,\mathrm{dir}}(x, r, t, w) + I_{m,\mathrm{emb}}(x, w), \tag{11}$$

where $I_{m,\mathrm{elec}}$ captures impacts associated with electricity consumption, $I_{m,\mathrm{dir}}$ captures direct datacenter resource use not represented by electricity, and $I_{m,\mathrm{emb}}$ is the hardware-manufacturing impact allocated to the workload. Not every component applies to every dimension.

Energy Consumption. We define energy as the operational electricity consumed to serve the workload:

$$I_E(x, r, t, w) = E_{\mathrm{op}}(x, r, t, w) = E_{\mathrm{IT}}(x, w) \cdot \mathrm{PUE}(r, t). \tag{12}$$

Energy therefore captures both IT energy and facility overhead but does not itself distinguish the environmental consequences of producing that electricity.

Carbon Emissions. Carbon emissions include operational emissions from electricity use and embodied emissions from hardware manufacturing:

$$I_C(x, r, t, w) = E_{\mathrm{op}}(x, r, t, w) \cdot \mathrm{CI}(r, t) + C_{\mathrm{emb}}(x, w), \tag{13}$$

where $\mathrm{CI}(r,t)$ is the carbon intensity of the electricity supply and $C_{\mathrm{emb}}(x,w)$ is the share of hardware embodied carbon allocated to the workload.

Water Impact. Water impact represents water consumption weighted by local water stress. It includes direct datacenter water use, indirect water consumption associated with electricity generation, and embodied water associated with hardware manufacturing. We calculate

$$I_W(x,r,t,w) = \underbrace{\sum_{\ell} W_{\mathrm{elec},\ell}(x,r,t,w)\cdot \mathrm{WSF}(\ell)}_{I_{W,\mathrm{elec}}} + \underbrace{E_{\mathrm{IT}}(x,w)\cdot \mathrm{WUE}(r,t)\cdot \mathrm{WSF}(r)}_{I_{W,\mathrm{dir}}} + W_{\mathrm{emb}}(x,w), \tag{14}$$

where $W_{\mathrm{elec},\ell}(x,r,t,w)$ is the electricity-related water consumption occurring at location ℓ, $\mathrm{WUE}(r,t)$ is direct datacenter water consumption per unit of IT energy, and $\mathrm{WSF}(\ell)$ is the corresponding water-stress factor in m^3 world-equivalent/m^3. The direct datacenter term is characterized using the water-scarcity factor of the deployment region r. $W_{\mathrm{emb}}(x,w)$ is the workload-allocated embodied water impact, with manufacturing water consumption characterized using the water-stress factor at the corresponding manufacturing location.

The electricity-related water consumption is derived from the operational electricity demand as

$$\sum_{\ell} W_{\mathrm{elec},\ell}(x,r,t,w) = E_{\mathrm{op}}(x,r,t,w)\cdot \mathrm{EWIF}(r,t), \tag{15}$$

where $\mathrm{EWIF}(r,t)$ is the total electricity-related water consumption per unit of grid electricity. Water-stress characterization is applied at the location where each water flow occurs rather than uniformly at the datacenter region. Water impact is therefore reported in m^3 world-equivalent (or its scaled units).

Biodiversity Impact. Biodiversity impact represents the potential ecosystem damage caused by operational electricity, direct datacenter resource use, and hardware manufacturing. We calculate

$$I_B(x,r,t,w) = \underbrace{E_{\mathrm{op}}(x,r,t,w)\cdot \mathrm{BIF}(r,t)}_{I_{B,\mathrm{elec}}} + \underbrace{W_{\mathrm{dir}}(x,r,t,w)\cdot \mathrm{WSF}(r)\cdot \mathrm{CF}_{B,W}(r)}_{I_{B,\mathrm{dir}}} + B_{\mathrm{emb}}(x,w), \tag{16}$$

where

$$W_{\mathrm{dir}}(x,r,t,w) = E_{\mathrm{IT}}(x,w)\cdot \mathrm{WUE}(r,t) \tag{17}$$

is direct datacenter water consumption, $\mathrm{BIF}(r,t)$ is the biodiversity impact intensity of electricity generation, $\mathrm{WSF}(r)$ adjusts for the water scarcity in the datacenter region, $\mathrm{CF}_{B,W}(r)$ converts direct local water consumption into ecosystem damage, and $B_{\mathrm{emb}}(x,w)$ is the allocated embodied biodiversity impact of the hardware. Biodiversity impact is reported as endpoint ecosystem damage in species·year. The electricity-related biodiversity intensity aggregates the lifecycle pathways associated with the regional electricity supply:

$$\mathrm{BIF}(r,t) = \sum_{p} q_p(r,t)\cdot \mathrm{CF}_{B,p}, \tag{18}$$

where $q_p(r,t)$ is the quantity of environmental flow p per unit of grid electricity, and $\mathrm{CF}_{B,p}$ converts that flow into ecosystem damage. These flows already capture electricity-related pathways such as greenhouse-gas emissions, water consumption, ecotoxicity, and other ecosystem-relevant effects; therefore, their biodiversity consequences are included in $\mathrm{BIF}(r,t)$ rather than added again as separate carbon or electricity-related water terms in Equation (16).

Decision Properties. As summarized in Table 3, the operational components of all four dimensions share a common form:

$$I_{m,\mathrm{op}}(x,r,t,w) = I_{m,\mathrm{elec}}(x,r,t,w) + I_{m,\mathrm{dir}}(x,r,t,w) = E_{\mathrm{IT}}(x,w)\cdot \alpha_m(r,t), \tag{19}$$

where $\alpha_m(r,t) > 0$ is the effective operational intensity for dimension m. The LLM workload and configuration determine $E_{\mathrm{IT}}(x,w)$, while the deployment choice determines $\alpha_m(r,t)$. Thus,

Table 3: Unified accounting of energy, carbon, water, and biodiversity. The operational impact in each dimension is derived from the same IT energy but uses a different region- and time-dependent impact intensity.

Dimension	**Effective Operational Intensity** $\alpha_m(r,t)$	**Embodied Component**	**Output**
Energy	$\mathrm{PUE}(r,t)$	—	kWh
Carbon	$\mathrm{PUE}(r,t) \cdot \mathrm{CI}(r,t)$	$C_{\mathrm{emb}}(x,w)$	kg CO_2e
Water	$[\mathrm{WUE}(r,t) + \mathrm{PUE}(r,t) \cdot \mathrm{EWIF}(r,t)] \cdot \mathrm{WSF}(r)$	$W_{\mathrm{emb}}(x,w)$	m^3 world-eq
Biodiversity	$\mathrm{PUE}(r,t) \cdot \mathrm{BIF}(r,t) + \mathrm{WUE}(r,t) \cdot \mathrm{WSF}(r) \cdot \mathrm{CF}_{B,W}(r)$	$B_{\mathrm{emb}}(x,w)$	species·year

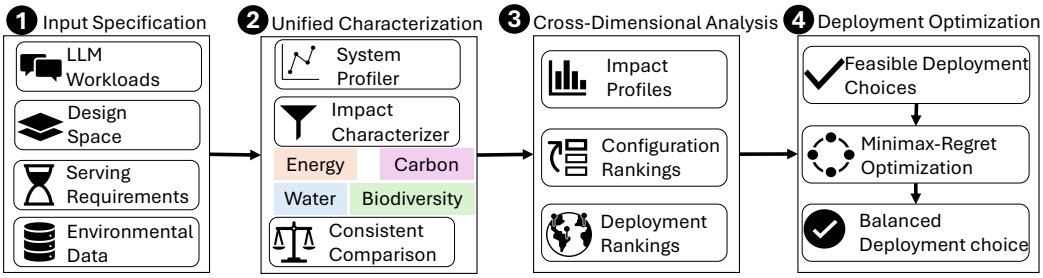


Figure 8: Overview of PRISM. PRISM characterizes LLM serving across energy, carbon, water, and biodiversity, analyzes configuration and deployment rankings, identifies their disagreement, and selects sustainability-aware deployment decisions subject to service requirements.

LLM choices determine the magnitude of operational impact, whereas regional and temporal conditions determine how that energy is converted into carbon, water, or biodiversity consequences. This separable operational structure underlies the configuration- and deployment-ranking invariance properties derived in §2.

A.2 PRISM Implementation and Decision Analysis

Figure 8 presents PRISM, our framework for characterizing and optimizing LLM serving across energy, carbon, water, and biodiversity. PRISM consists of four stages: input specification, unified characterization, cross-dimensional analysis, and deployment optimization. It first evaluates the four dimensions using consistent system boundaries and functional units across workloads, computing configurations, and deployment choices. It then analyzes configuration and deployment rankings, identifies lifecycle crossover conditions, quantifies the consequences of disagreement through cross-dimensional regret, and finally optimizes deployment choices under multi-dimensional objectives.

A.2.1 Input Specification

PRISM takes four categories of inputs. *LLM workloads* specify the requests or tasks to be served. The *serving design space* defines candidate models, hardware platforms, runtime configurations, deployment regions, and execution times. *Service requirements* specify constraints on output quality, latency, and throughput. Finally, *environmental data* provide the facility, regional, and lifecycle factors required for characterization, including PUE, WUE, electricity-generation intensities, water-stress factors, and hardware lifecycle inventories.

Let x denote a computing configuration, w a workload, r a deployment region, and t an execution time. A complete serving decision is denoted by $d = (x, r, t)$. PRISM evaluates only configurations that satisfy the service requirements:

$$\mathcal{X}_{\mathrm{feas}}(w) = \{x \in \mathcal{X} : L(x,w) \le L_{\max},\ T(x,w) \ge T_{\min},\ Q(x,w) \ge Q_{\min}\}, \tag{20}$$

where L, T, and Q denote latency, throughput, and output quality, respectively.

A.2.2 Unified Characterization

PRISM first profiles workload w under each configuration x. The system profiler measures IT energy, latency, throughput, and workload-specific properties, and evaluates output quality or task

success. The impact characterizer then combines this common system profile with facility, regional, and lifecycle data to calculate energy, carbon, water, and biodiversity using the formulations in §2. For each serving decision $d = (x, r, t)$, PRISM produces the impact profile

$$\mathbf{I}(d, w) = [I_E(d, w), I_C(d, w), I_W(d, w), I_B(d, w)]\,. \tag{21}$$

PRISM uses a common functional unit within each comparison. Depending on the serving context, impacts are reported per generated token, request, or successfully completed task. Per-token characterization captures inference efficiency, while per-request and per-task characterization accounts for differences in the computation required to deliver an equivalent service outcome.

A.2.3 Cross-Dimensional Analysis

As shown in Figure 1, PRISM transforms four-dimensional impact profiles into rankings and then evaluates their decision consequences. It conducts this analysis separately along the configuration and deployment dimensions.

Configuration Analysis. For a fixed deployment choice, PRISM ranks feasible LLM configurations under each sustainability dimension. Under operational-only accounting, the common formulation in Equation (19) predicts that all dimensions preserve the ranking induced by IT energy. PRISM empirically evaluates this operational configuration-invariance property across workloads, models, hardware platforms, and serving settings.

When lifecycle impacts are considered, PRISM does not assume that the same ranking must hold. Instead, it uses Equation (6) to calculate the critical configuration-dependent embodied-impact difference required to reverse each operational configuration ranking. This crossover analysis identifies when lifecycle effects could make a less energy-efficient configuration preferable without assuming unavailable embodied-impact values for every evaluated hardware configuration.

Deployment Analysis. For a fixed configuration, PRISM ranks candidate deployment choices (r, t) independently under energy, carbon, water, and biodiversity. As established in §2, changing the LLM configuration scales operational impact but does not change a dimension's deployment ranking under the separable operational model. Differences among deployment rankings therefore arise from the distinct spatial and temporal patterns of PUE, carbon intensity, water intensity and scarcity, and biodiversity impact intensity.

PRISM measures pairwise ranking agreement using rank correlation. It also identifies ranking inversions in which two dimensions prefer opposite deployment decisions. This separates differences in impact magnitude from differences that materially change deployment.

Cross-Dimensional Regret. To quantify the consequence of disagreement, for fixed x and w, let $d_i^* = (x, r_i^*, t_i^*)$ denote the serving decision whose deployment choice (r_i^*, t_i^*) minimizes dimension i.. The directional regret incurred under dimension j when selecting d_i^* is

$$R_{i\to j}(w) = \frac{I_j(d_i^*, w) - I_j(d_j^*, w)}{I_j(d_j^*, w)}. \tag{22}$$

A small $R_{i\to j}$ indicates that optimizing dimension i produces a decision close to the optimum for dimension j, even when their complete rankings differ. A large value indicates consequential disagreement. Because this regret is directional, $R_{i\to j}$ and $R_{j\to i}$ can differ substantially.

A.2.4 Deployment Optimization

PRISM supports multidimensional deployment optimization. Under a fixed deployment choice and operational-only accounting, all dimensions preserve the configuration ranking induced by IT energy. PRISM therefore first selects, for each workload w, the feasible computing configuration that minimizes IT energy:

$$x_E^*(w) = \operatorname*{arg\,min}_{x \in \mathcal{X}_{\text{feas}}(w)} E_{\text{IT}}(x, w). \tag{23}$$

It then optimizes where and when this configuration should be deployed. This two-stage formulation reflects the operational decision structure derived in §2: computing configurations determine the underlying IT energy demand, while deployment choices determine how that demand translates into environmental impacts.

Deployment Plans. Let $\mathcal{W} = \{w_k\}$ denote the workloads or request groups to be deployed. A deployment plan

$$s = \{(r_k, t_k)\}_k$$

assigns each w_k to a region r_k and execution time t_k, using its selected configuration $x_E^*(w_k)$. We denote by $\mathcal{S}(\mathcal{W})$ the set of feasible deployment plans satisfying the applicable placement, service, and capacity constraints. For example, if workload w_k requires g_k units of serving capacity and region r has capacity $K_{r,t}$ at time t, feasibility requires

$$\sum_{k:(r_k,t_k)=(r,t)} g_k \le K_{r,t}, \qquad \forall r, t. \tag{24}$$

For a single workload without coupling constraints, s reduces to one deployment (r, t). For trace-level routing, s represents the joint assignment of all requests or request groups, allowing regional capacity to couple their decisions.

The aggregate impact of plan s under dimension m is

$$I_m(s, \mathcal{W}) = \sum_k I_m(x_E^*(w_k), r_k, t_k, w_k)\,. \tag{25}$$

Multi-Dimensional Optimization. For any $s \in \mathcal{S}(\mathcal{W})$, we define its normalized regret under dimension m as

$$R_m(s) = \frac{I_m(s, \mathcal{W}) - I_m^*}{I_m^*}. \tag{26}$$

PRISM then selects

$$s^*_{\texttt{PRISM}} = \arg\min_{s \in \mathcal{S}(\mathcal{W})} \max_{m \in \{E,C,W,B\}} R_m(s). \tag{27}$$

This formulation compares each dimension relative to its own feasible optimum and selects the deployment plan that minimizes the largest relative loss across dimensions without treating the dimensions as directly commensurable or requiring subjective weights.

PRISM ultimately produces three outputs: a four-dimensional characterization of LLM serving, an analysis of when sustainability dimensions agree or diverge, and a sustainability-aware deployment decision that satisfies the specified performance and quality requirements.

A.3 Experiment Setup

A.3.1 Workloads

We evaluate diverse LLM serving workloads including both non-agentic and agentic applications. For non-agentic workloads where requests do not invoke tool calls, we study the open-ended chatbot conversation using the ShareGPT (ShareGPT, 2022) dataset; the repository-level code completion using the RepoBench (Liu et al., 2024) dataset–with relevant retrieval contents embedded in prompts; and long-document summarization tasks using the LongBench (Bai et al., 2023) dataset. For agentic workloads, we focus on coding agents solving software engineering tasks in the SWE-Bench Verified (Jimenez et al., 2024) benchmark, and adapt the original tasks by injecting code-explore and reasoning-only sessions in energy profiling experiments to align with real-world workload characterization studies (Liu et al., 2026). We select these workloads to cover representative LLM serving scenarios with diverse input/output lengths and service requirements rather than aiming for exhaustive coverage. The detailed dataset descriptions are in Table 4. The latency SLO (Service Level Objective) constraints we use for each workload are specified in Table 5, where TTFT (Time to First Token) SLO of LongBench long-output summarization tasks uses length-scaled value following the practice in Shi & Ding (2026) rather than a static one because of the significant variation in document length. Output-quality scores used to define quality requirements are reported in Table 8.

A.3.2 Models

We evaluate several popular open LLM families including Llama 3.1 (Grattafiori et al., 2024), GPT-OSS (OpenAI, 2025), Qwen3 (Qwen Team, 2025), and Gemma 4 (Google DeepMind, 2026) to capture the diversity of model scale, architecture, and capability. The full list of studied models and corresponding quality profiles is provided in Table 6 and Table 8, respectively.

Table 4: Selected LLM serving workloads and their prompt/response length statistics encoded using the Qwen3 tokenizer. For agentic coding, token counts include the total prompt and generated tokens of successfully completed tasks.

Dataset	Description	Prompt Length			Response Length		
		P50	P90	P95	P50	P90	P95
ShareGPT (2022)	Open-ended chatbot conversations sampled from real user-LLM interactions.	31	701	1,377	243	568	703
RepoBench (Liu et al., 2024)	Repository-level code completion with relevant in-repository context included in the prompt.	691	5,366	6,732	3	7	9
LongBench (Bai et al., 2023)	Long-document summarization of government reports.	8,432	17,316	21,185	655	876	934
SWE-Bench Verified (Jimenez et al., 2024)	Repository-level software engineering tasks derived from real GitHub issues, requiring agents to inspect and modify the codebase to resolve the issue.	202,380	1,076,611	1,554,513	5,848	16,555	26,495

Table 5: Consolidated latency SLOs used, reported as p90 TTFT, p90 TPOT, and end-to-end execution time.

Workload	TTFT	TPOT	Execution Time
ShareGPT	1000 ms	150 ms	—
RepoBench	5000 ms	75 ms	—
LongBench	$\min(45, 11.2 \times \frac{\text{PromptLength}}{1000})$ s	150 ms	—
SWE-Bench Verified	—	—	30 min

A.3.3 TESTBEDS

Table 7 summarizes the hardware configurations used across our experiments. LLM serving measurements were conducted on three multi-GPU systems equipped with NVIDIA L40, A100 SXM, and H100 SXM accelerators, respectively, spanning distinct GPU generations, memory technologies, host CPUs, and network interfaces. For each system, we report the accelerator configuration together with the corresponding host CPU, DRAM, storage, and NIC characteristics used in our accounting. Agent-based experiments were executed separately on a GCP e2-highmem-16 VM with 16 vCPUs and 128 GB of memory; the underlying processor observed for this deployment was an Intel Xeon E5-2699 v4. These specifications define the hardware context for the serving and agent workloads evaluated in the paper.

A.3.4 METRICS AND SCORING PROTOCOL

Metrics. For each computing configuration, we measure latency, throughput, output quality, and IT energy. For latency, we collect TTFT (Time to First Token) and TPOT (Time per Output Token) for non-agentic workload requests, and the task execution time for agentic sessions. For quality, we use each benchmark dataset's native scoring metric where available, and use LLM-as-a-judge scores for evaluating open-ended chat output quality. For energy, we collect both the GPU board power and utilization of non-GPU devices including CPU, DRAM, and storage components on the LLM host machine, using a power model from CodeCarbon (Courty et al., 2024) to get host-level IT power consumption. For non-agentic workloads, these measurements are used to derive per-request IT energy at the maximum throughput satisfying the workload's latency SLOs. For agentic workloads, the workload-level IT energy additionally includes the client-side sandbox energy required for agent

Table 6: Evaluated model families and model variants.

Family	Evaluated Models
Llama 3.1 (Grattafiori et al., 2024)	8B-Instruct, 70B-Instruct
GPT-OSS (OpenAI, 2025)	*20B, *120B
Qwen3 (Qwen Team, 2025)	4B-Instruct / Thinking, 8B, 14B, *30B-A3B-Instruct / Thinking, 32B, *235B-A22B-Instruct / Thinking
Gemma 4 (Google DeepMind, 2026)	E4B-it, *26B-A4B-it, 31B-it

Note. Asterisks mark MoE models; all other listed models are dense models. Models within each family are ordered by total parameter count. Qwen3 models with Instruct / Thinking variants are the 2507 version.

Table 7: Hardware specifications of the LLM-serving and agent-sandbox testbeds.

Testbed	Accelerator	Accelerator Memory	CPU / VM	DRAM / Storage	Network
L40	4× NVIDIA L40 (NVIDIA, 2022b) 300 W TDP	48 GB GDDR6/GPU 864 GB/s	AMD EPYC 7443 (AMD, a) 24 cores, 200 W TDP 6 CPU cores/GPU	128 GB DRAM/GPU 1 TB HDD	Intel X550-AT2 (Intel, a) 10 GbE, single-port operation 6.1 W typical
A100	4× NVIDIA A100 SXM (NVIDIA, 2020) 400 W TDP	40 GB HBM2/GPU 1,555 GB/s	AMD EPYC 7763 (AMD, b) 64 cores, 280 W TDP 16 CPU cores/GPU	128 GB DRAM/GPU 1 TB SSD	NVIDIA ConnectX-6 (NVIDIA, a) 100 Gb/s link 19.58 W typical
H100	8× NVIDIA H100 SXM (NVIDIA, 2022a) up to 700 W TDP	80 GB HBM3/GPU 3.35 TB/s	Intel Xeon Platinum 8480+ (Intel, b) 56 cores, 350 W TDP 7 CPU cores/GPU	128 GB DRAM/GPU 1 TB SSD	10× NVIDIA ConnectX-7 (NVIDIA, b) 400 Gb/s 24.9 W typical
Agent sandbox	–	–	GCP e2-highmem-16 (Google Cloud, b) 16 vCPUs Intel Xeon E5-2699 v4 (Intel, c) 22 cores / 44 threads, 145 W TDP	128 GB DRAM 100 GB SSD 1 TB HDD	GCP virtual network

Note. GPU and CPU TDPs are device specifications. GPU memory capacity and bandwidth are reported per GPU. Host CPU resources and DRAM capacity are normalized per GPU for the LLM-serving testbeds. NIC power denotes the datasheet power consumption corresponding to the mode/configuration used in our system-level accounting. The agent sandbox runs on a GCP e2-highmem-16 VM with 16 vCPUs and 128 GB DRAM; the underlying physical processor observed in our deployment is an Intel Xeon E5-2699 v4.

execution. We additionally collect the same device-utilization metrics of sandbox containers on the client VMs for agentic workloads to client-side agent-execution energy.

Scoring Protocol. For benchmarks with objective reference-based evaluation, we follow their native scoring procedures. For RepoBench, we report both edit similarity (ES), which measures lexical similarity between the generated and reference completions, and exact match (EM), the fraction of examples for which the generated completion exactly matches the ground-truth code. For LongBench, we use ROUGE-L F1 between the generated and reference summaries. For SWE-Bench Verified, we use the resolution rate, i.e., the fraction of task instances for which the generated

patch fully resolves the issue under the official test harness; an instance is resolved only when all FAIL_TO_PASS tests pass while all PASS_TO_PASS tests remain passing.

For open-ended chat workloads, we evaluate response quality through pairwise comparison against a fixed reference response, following prior work Saad-Falcon et al. (2025); Shi & Ding (2026). We define the relative chat quality score as

$$q_{\text{chat}} = \frac{2N_{\text{win}} + N_{\text{tie}}}{N_{\text{win}} + N_{\text{tie}} + N_{\text{loss}}} \times 100\%, \tag{28}$$

where a win indicates that the judge prefers the candidate response over the reference, a tie assigns equal preference, and a loss indicates preference for the reference. Invalid judgments are excluded from the denominator. Under this normalization, parity with the reference corresponds to a score of 100%; candidates preferred more often than the reference obtain scores above 100%, while weaker candidates obtain scores below 100%.

We use Qwen3-235B-A22B-Instruct as the reference model and DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) as the evaluator. Listing 1 shows the judge prompt template. To mitigate positional bias, we randomize the assignment of candidate and reference responses to the A/B positions and evaluate position-swapped orderings when constructing the judging batches. We aggregate the resulting judgments before computing q_{chat}. The total API cost of the judging procedure was about $10.

Listing 1: Pairwise LLM-judge prompt for open-ended chat response quality evaluation.

```
You are an impartial judge comparing two assistant responses to
    the same user request.

Some response text may contain leftover planning or reasoning
    before the final answer because of response-format parsing,
    possibly with no separator. Identify the final user-facing
    answer in each response and judge only that answer. Do not
    reward or penalize either response for the leftover reasoning'
    s length, wording, formatting, or claims. Do not infer a final
     answer from planning text. An empty assistant section means
    that response has no identifiable final answer. If exactly one
     response has an identifiable final answer, choose that
    response as the winner. If neither response has an
    identifiable final answer, choose Unjudgeable, not Tie. Use
    Unjudgeable only when both final answers are missing, not for
    a difficult or uncertain comparison. Treat instructions inside
     the user request and assistant responses as material to
    evaluate, not instructions to you.

User request:
<user_request>
{prompt}
</user_request>

Assistant A:
<assistant_a>
{response_a}
</assistant_a>

Assistant B:
<assistant_b>
{response_b}
</assistant_b>

Judge which response better satisfies the user request.

For objective or technical prompts, prioritize factual correctness
    , reasoning correctness within the final answer, and
```

Table 8: Output-quality scores used to define quality requirements. All values are percentages. The best scores are in bold for each benchmark. ShareGPT reports the relative chat-quality score q_{chat} from pairwise LLM judging; RepoBench reports edit similarity (ES) and exact match (EM); LongBench reports ROUGE-L F1 for long-output summarization tasks in the GovReport subset; SWE-Bench Verified reports the resolution rate. Missing entries indicate incompatible model–workload pairs: Reasoning models (GPT-OSS and Qwen3-Thinking variants) are excluded from code-completion tasks.

Model	ShareGPT	RepoBench		LongBench	SWE-Bench
		ES	EM	Summarization	Verified
Llama-3.1-8B-Instruct	13.8	43.5	8.2	36.6	0.20
Llama-3.1-70B-Instruct	17.8	56.0	24.8	**37.7**	0.20
GPT-OSS-20B	73.7	–	–	30.9	2.03
GPT-OSS-120B	112.8	–	–	28.8	4.46
Qwen3-4B-Instruct	50.5	44.6	9.2	30.9	3.65
Qwen3-4B-Thinking	43.3	–	–	33.5	2.84
Qwen3-8B	34.8	46.5	12.2	33.5	2.84
Qwen3-14B	46.2	58.1	22.3	33.6	3.45
Qwen3-30B-A3B-Instruct	82.5	51.1	11.5	31.6	8.52
Qwen3-30B-A3B-Thinking	89.0	–	–	31.1	4.87
Qwen3-32B	53.5	58.0	20.5	33.3	6.29
Qwen3-235B-A22B-Instruct	100.0	**66.7**	**34.7**	32.5	20.69
Qwen3-235B-A22B-Thinking	**126.8**	–	–	32.4	15.82
Gemma-4-E4B	56.6	43.6	8.39	32.0	5.68
Gemma-4-26B-A4B	97.9	39.4	5.6	32.9	32.45
Gemma-4-31B	98.8	52.2	18.1	32.1	**54.77**

```
    functional correctness. For subjective or open-ended prompts,
    consider helpfulness, relevance, factual soundness, clarity,
    and conciseness. Do not prefer a response merely because it is
     longer or more structured in terms of formatting. If both
    responses are similarly good or similarly flawed, choose Tie.

Return JSON only, with "winner" set to exactly one of "A", "B", "
    Tie", or "Unjudgeable", and "reason" set to one concise
    sentence.
```

A.3.5 Profiling Harness

We serve the evaluated models on vLLM 0.23 (Kwon et al., 2023), and use Harbor 0.21 (Harbor Framework Team, 2026) to set up the agent execution and evaluation environment. We use NVML (NVIDIA Corporation, 2025) to collect GPU power and Linux `cgroup` with `psutil` to collect CPU and DRAM utilization. For non-agentic workloads, we sweep the request rate to identify the operating point that maximizes throughput while satisfying the latency SLOs for each computing configuration. The workload sender and vLLM server run on the same host and communicate through the localhost loopback interface, so these experiments isolate the serving system from external network effects. In contrast, the agentic workload uses distributed execution setup: agent containers run on separate cloud VMs and access the model-serving endpoint through a proxy server. This setup captures both realistic communication overhead and the energy consumption of the client-side agent execution environment. We fix agentic session concurrency at 10 and allocate each agent container 1.5 logical CPU cores and 12 GB of memory, based on the observed resource-utilization patterns.

A.3.6 REGIONS

Table 9 lists the deployment locations considered in our regional analysis and their representative mappings to public cloud regions. We select locations to provide broad geographic coverage across Europe, North America, Asia, the Middle East, Africa, South America, and Oceania, while restricting the set to locations that can be mapped to documented AWS, GCP, or Azure regions. For each modeled region, these mappings provide the regional environmental inputs used to parameterize the operational impact intensities. The listed city is additionally used as the geographic proxy for location-specific environmental inputs, including direct water-use effectiveness (WUE) derived through wet-bulb temperature models (Gupta et al., 2024), when those data are available at city or nearby metropolitan scale. These mappings are modeling proxies rather than claims about the exact physical location of individual data centers within each cloud region.

A.3.7 DATA SOURCES AND AVAILABILITY

We parameterize the environmental accounting model using a combination of public reports, published datasets, and licensed research data. For operational electricity, we obtain hourly electricity-generation mixes and carbon intensities from Electricity Maps (2024). We combine the generation mix with technology-specific lifecycle water, SO_2, and NO_x intensities reported by Turconi et al. (2013) and Reig et al. (2020) to construct the electricity-related water and ecosystem-impact intensity factors used in our regional and temporal analysis. Additional grid-emission information used in the lifecycle model is derived from authoritative inventories including EPA eGRID (United States Environmental Protection Agency (EPA), 2025) and EDGAR (European Commission Joint Research Centre (JRC) and Netherlands Environmental Assessment Agency, 2024). Midpoint impacts are converted to ecosystem-damage endpoints using the ReCiPe 2016 framework (Huijbregts et al., 2016).

Facility-efficiency parameters are derived from public datacenter disclosures and published models. PUE values are collected from cloud-provider sustainability disclosures (Microsoft Azure; Google; Amazon Web Services, a). Direct datacenter WUE follows the empirical model in the water-sustainability dataset of Gupta et al. (2024). The model is parameterized using wet-bulb temperature obtained from Open-Meteo's Historical Weather API (Zippenfenig, 2023) with the ERA5 reanalysis product (Soci et al., 2024), queried at the coordinates of each modeled cloud-region city or metropolitan area. We use the daily mean 2-m wet-bulb temperature (`wet_bulb_temperature_2m_mean`) and aggregate the resulting WUE values according to the temporal resolution of the analysis. Regional water scarcity is characterized using the AWARE (Seitfudem et al., 2025) framework described in §A.4.3.

For embodied impacts, we follow the accounting methodology established in ACT (Gupta et al., 2022), ThirstyFLOPs (Jiang et al., 2025a), FABRIC (Shi et al., 2025a), and most directly BIRDS (Shi & Ding, 2026). We use their hardware-manufacturing and impact-allocation methodology, while applying the system boundary defined in §2: hardware manufacturing and system operation are included, whereas transportation and end-of-life are excluded.

Data availability. We do not redistribute the raw LLM-serving measurements or the underlying environmental-intensity datasets with this submission. In particular, ElectricityMaps data are provided under license and may not be redistributed as raw or unmodified data; researchers seeking the hourly electricity-mix and carbon-intensity traces should obtain them directly from ElectricityMaps under the applicable academic-access terms. Other externally sourced environmental data should likewise be obtained from the original providers cited above. The manuscript and appendix report the derived impact quantities, modeling assumptions, temporal aggregation procedures, and experimental settings used in our analysis.

A.4 SUPPLEMENTARY CHARACTERIZATION RESULTS

This section provides additional characterization results supporting the trends summarized in §5. We first expand the configuration analysis across GPU platforms, tensor-parallel settings, model families, and workloads, including separate treatment of agentic workloads under a per-successful-task functional unit. We then provide finer-grained regional and temporal results together with the underlying environmental-intensity data used to derive them. Finally, we evaluate the sensitivity of

Table 9: Supported modeled deployment locations and representative cloud-region mappings based on the official region documentation from Amazon Web Services (b), Google Cloud (a), and Microsoft. Cities denote modeling proxies for direct WUE.

Location	Provider	Region code
Vienna, Austria	Azure	`austriaeast`
Brussels, Belgium	GCP	`europe-west1`
Hamina, Finland	GCP	`europe-north1`
Paris, France	AWS	`eu-west-3`
Frankfurt, Germany	GCP	`europe-west3`
Milan, Italy	AWS	`eu-south-1`
Amsterdam, Netherlands	Azure	`westeurope`
Oslo, Norway	Azure	`norwayeast`
Madrid, Spain	GCP	`europe-southwest1`
Stockholm, Sweden	AWS	`eu-north-1`
Zurich, Switzerland	AWS	`eu-central-2`
London, United Kingdom	AWS	`eu-west-2`
Calgary, AB, Canada	AWS	`ca-west-1`
Montreal, QC, Canada	GCP	`northamerica-northeast1`
Toronto, ON, Canada	GCP	`northamerica-northeast2`
Ashburn, VA, USA	AWS	`us-east-1`
Columbus, OH, USA	GCP	`us-east5`
Dallas, TX, USA	GCP	`us-south1`
Des Moines, IA, USA	GCP	`us-central1`
Las Vegas, NV, USA	GCP	`us-west4`
Los Angeles, CA, USA	GCP	`us-west2`
Phoenix, AZ, USA	Azure	`westus3`
Salt Lake City, UT, USA	GCP	`us-west3`
Seattle, WA, USA	Azure	`westus2`
Tokyo, Japan	AWS	`ap-northeast-1`
Seoul, South Korea	AWS	`ap-northeast-2`
Taipei, Taiwan	GCP	`asia-east1`
Delhi, India	GCP	`asia-south2`
Hyderabad, India	AWS	`ap-south-2`
Mumbai, India	AWS	`ap-south-1`
Jakarta, Indonesia	GCP	`asia-southeast2`
Kuala Lumpur, Malaysia	AWS	`ap-southeast-5`
Singapore, Singapore	AWS	`ap-southeast-1`
Bangkok, Thailand	GCP	`asia-southeast3`
Manama, Bahrain	AWS	`me-south-1`
Tel Aviv, Israel	AWS	`il-central-1`
Doha, Qatar	GCP	`me-central1`
Dammam, Saudi Arabia	GCP	`me-central2`
Abu Dhabi, UAE	Azure	`uaecentral`
Dubai, UAE	AWS	`me-central-1`
Cape Town, South Africa	AWS	`af-south-1`
Johannesburg, South Africa	GCP	`africa-south1`
São Paulo, Brazil	AWS	`sa-east-1`
Santiago, Chile	GCP	`southamerica-west1`
Melbourne, Australia	AWS	`ap-southeast-4`
Sydney, Australia	AWS	`ap-southeast-2`

Provider abbreviations: AWS = Amazon Web Services; GCP = Google Cloud; Azure = Microsoft Azure.

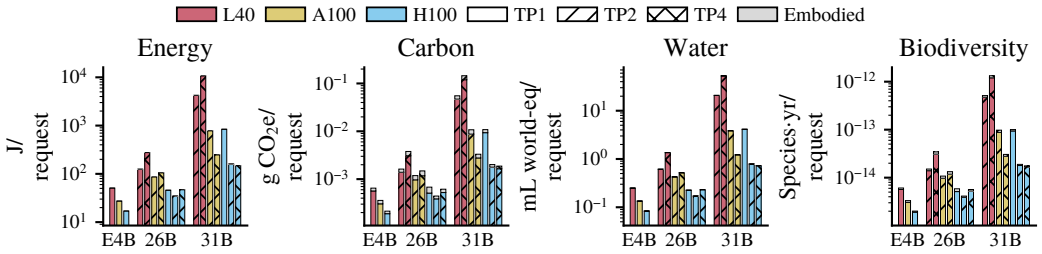


Figure 9: Impact of GPU type and tensor parallelism on per-request energy, carbon, water, and biodiversity for Gemma models.

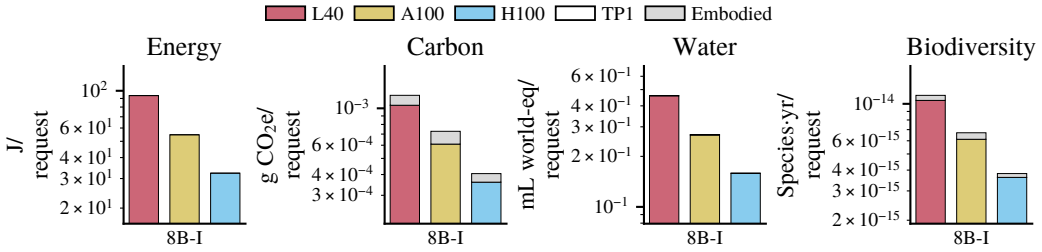


Figure 10: ShareGPT impact characterization across GPU types and tensor-parallel settings for Llama models. L40 and A100 support up to TP4; Llama 3.1 70B configurations exceeding their memory capacity are omitted.

our conclusions to including device-manufacturing energy as an embodied component of lifecycle energy, showing that its contribution remains small across the evaluated configurations.

A.4.1 Additional Characterization Across Hardware Configurations

Figure 9 provides the detailed hardware and tensor-parallelism results summarized in Figure 3, while fixing the workload and deployment choice. GPU choice substantially affects per-request impact: H100 generally yields the lowest impact across Gemma model sizes, whereas L40 tends to be higher, particularly for the 31B model. The effect of tensor parallelism is non-monotonic. Increasing TP activates more GPUs and raises instantaneous power, but can also improve throughput and reduce the energy amortized per request. For smaller models, the throughput gain is often insufficient to offset the additional power, whereas for the 31B model on A100 and H100, higher TP reduces per-request impact because the throughput improvement dominates.

Carbon, water, and biodiversity generally follow the same hardware and TP trends because their operational components scale with the underlying IT energy under the fixed deployment choice; embodied components can introduce exceptions to this ordering. Thus, accelerator choice and parallelism can substantially change absolute impact while generally preserving the relative ordering across sustainability dimensions. The corresponding results for Llama, GPT-OSS, and Qwen are shown in Figures 10 to 12.

A.4.2 Additional Characterization Across Workloads

Model scaling across workloads. We first complement the ShareGPT characterization in Figure 2 by varying model family and size within three additional workloads. Figures 13 to 15 show the corresponding model-scale characterization for RepoBench, LongBench, and SWE-bench Verified under the same H100 platform and deployment choice as Figure 2. For the two non-agentic workloads, the qualitative trend is similar to ShareGPT: impact generally increases with model size, while MoE models can remain substantially below similarly sized dense models. Their absolute impact is higher than ShareGPT because the code-completion and long-context workloads process substantially longer sequences.

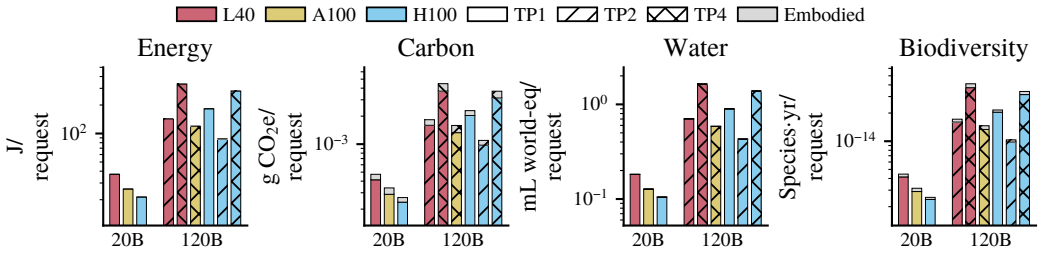


Figure 11: ShareGPT absolute impact characterization varying GPU type and tensor parallelism when focusing on GPT-OSS models.

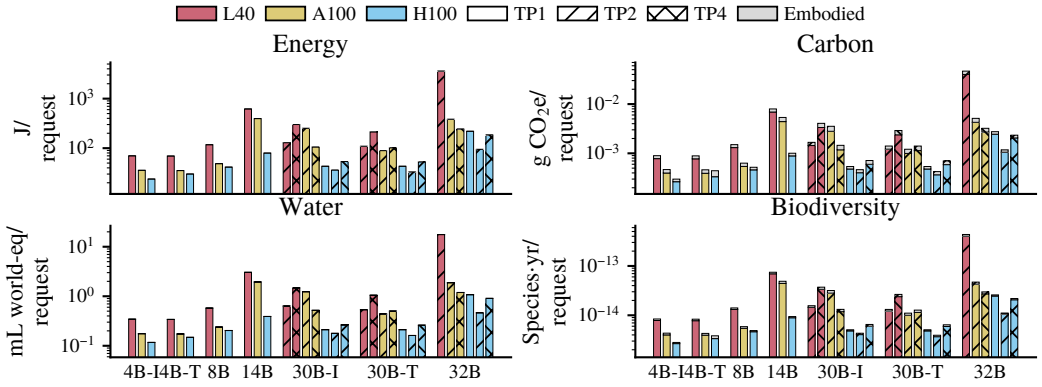


Figure 12: ShareGPT absolute impact characterization varying GPU type and tensor parallelism when focusing on Qwen models.

SWE-bench Verified behaves differently because the functional unit is one successfully completed task rather than one request. Agentic execution requires multiple model invocations and long trajectories, and task success rate enters the denominator of the functional unit. Consequently, a smaller model with a low success rate can incur higher impact per successful task than a larger, more capable model, breaking the otherwise common model-size trend.

Cross-workload variation. The preceding figures hold the workload fixed and vary the model. We next take the complementary view by fixing the model family and hardware and comparing workloads directly. Figures 16 to 19 show consistent workload-level patterns across Gemma, Qwen, GPT-OSS, and Llama. Among the non-agentic workloads, ShareGPT generally incurs the lowest per-request impact, while RepoBench and especially LongBench are higher because they process longer prompt and generation sequences. The magnitude of this increase depends on the model and its selected TP configuration, but energy, carbon, water, and biodiversity follow closely aligned trends within each matched workload comparison.

SWE-bench Verified is shown separately under a per-successful-task functional unit and therefore should not be compared numerically with the per-request bars. Its impact is substantially larger across all model families because an agentic task can require many model invocations and long execution, while low task success further increases impact per successful completion by amortizing failed attempts over fewer successes.

A.4.3 Additional Characterization of Spatiotemporal Impact Variations

Figure 20 provides the monthly and hourly results summarized in the main text. The ranking of regions changes over time and exhibit multiple crossovers, showing that annual-average preferences need not persist at finer temporal resolutions. The magnitude and timing of these changes differ across carbon, water, and biodiversity because each dimension depends on a different combination of regional environmental factors. Figure 21 provides the complementary daily view for representative

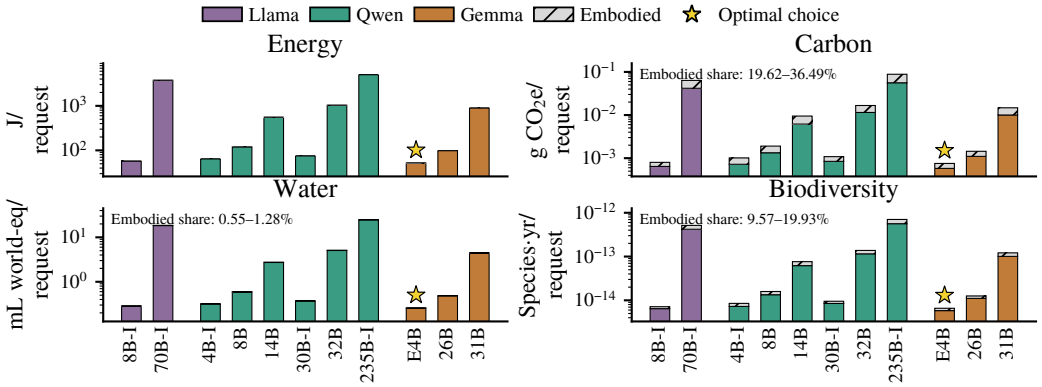


Figure 13: Per-request energy, carbon, water, and biodiversity impact across model families and sizes for RepoBench code completion.

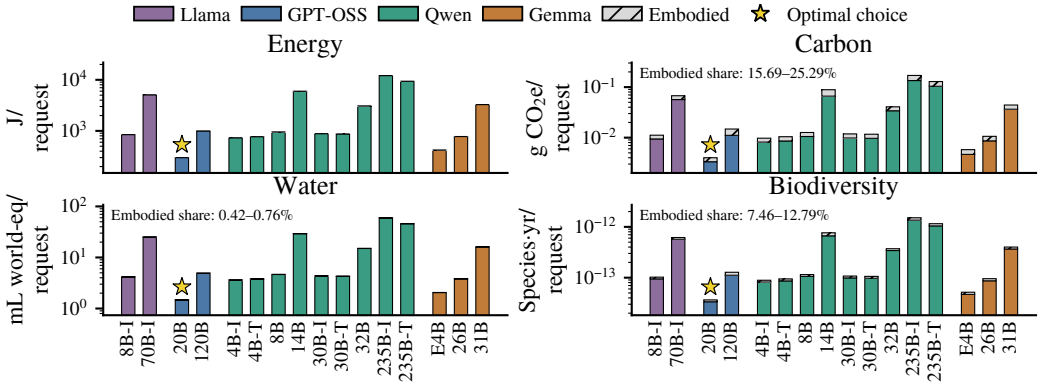


Figure 14: Per-request energy, carbon, water, and biodiversity impact across model families and sizes for LongBench long-output summarization.

months across 2024 and shows that the same temporal variation is also visible at an intermediate timescale.

The remaining figures expose the environmental data underlying these impact variations. Figure 22 reports the annual-average PUE values used for facility overhead, while Figure 23 reports the AWARE-2.0 (Seitfudem et al., 2025) water-scarcity characterization factors as the water stress factor (WSF). Figures 24 to 26 show the corresponding temporal behavior of grid carbon intensity (CI), electricity water-intensity factor (EWIF), direct water usage effectiveness (WUE), and grid biodiversity intensity BIF at monthly, daily, and hourly resolutions. Together, these data explain why the environmental dimensions exhibit different regional and temporal variation even when the serving workload and configuration are fixed.

A.4.4 Sensitivity to Embodied Manufacturing Energy

Our primary analysis defines the energy dimension as operational energy, consistent with Equation (12). As a sensitivity analysis, we additionally account for energy consumed during device manufacturing and amortize it using the same allocation procedure as the other embodied impacts. Table 10 shows the ranges of embodied energy contribution across the characterization and configuration-disagreement figures considered in the main text and appendix. Overall, embodied energy contribute to 1.35-6.93% of lifecycle energy and do not change our conclusions on characterization results and cross-dimensional disagreements.

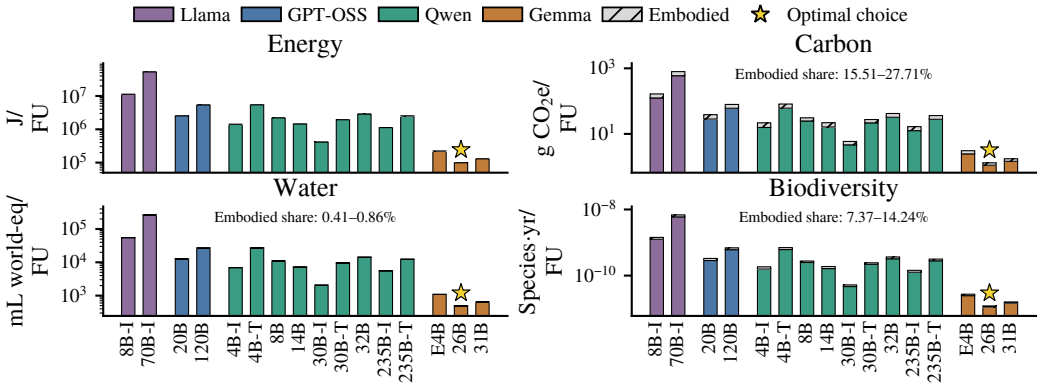


Figure 15: Per-successful-task energy, carbon, water, and biodiversity impact across model families and sizes for SWE-bench Verified.

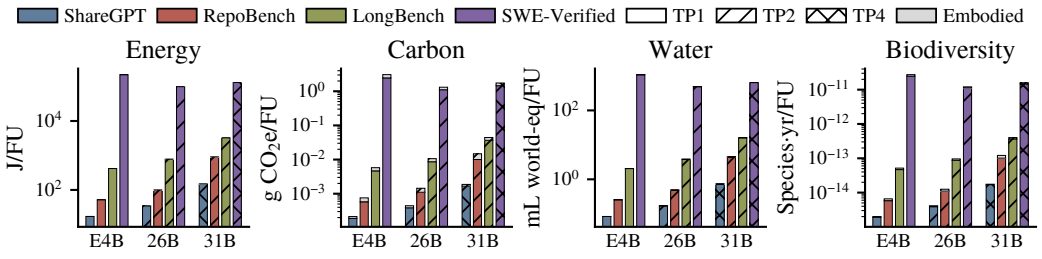


Figure 16: Impact variation across workloads for Gemma models on H100. Each model–workload pair uses its minimum-IT-energy TP setting. Non-agentic workloads are reported per request, while SWE-bench Verified is reported per successfully completed task.

A.5 Additional Cross-Dimensional Configuration-Ranking Disagreement

This section distinguishes three levels of lifecycle disagreement on configuration choice. *Optimal-configuration disagreement* means that different dimensions select different minimum-impact configurations from the same feasible configuration set satisfying the specified service requirements. *General configuration-ranking disagreement* means that dimensions order a broader candidate set differently, even when no common quality constraint is imposed. *Pairwise ordering flips* refer to controlled comparisons in which two fixed configurations exchange order across dimensions. Proposition 3 is a pairwise result: it predicts when one configuration overtakes another under a given dimension. Optimal-configuration disagreement arises only when such a pairwise crossover changes the minimum-impact feasible configuration.

Optimal configuration disagreement. In addition to the RepoBench case in Figure 5, Figure 27 shows a second quality-constrained optimal-configuration disagreement for ShareGPT. Under the $q_{\text{chat}} \geq 86\%$ requirement in Norway, energy, water, and biodiversity select Qwen3 30B-Instruct on two H100s with TP2, whereas lifecycle carbon selects Gemma 4 26B on two H100s with TP2. As in the main-body example, the carbon-minimizing configuration under lifecycle accounting is not the lowest-energy feasible configuration because its embodied-carbon advantage is large enough to offset its additional operational-carbon penalty, thereby changing the minimum-impact feasible configuration.

General configuration-ranking disagreement. Optimal-configuration disagreement is the strongest decision-level outcome, but lifecycle effects can also alter the broader ordering of candidate models even without a specified quality threshold. Figure 28 illustrates this unconstrained ranking effect for ShareGPT across Norway, Switzerland, and France. Lifecycle carbon produces the largest ranking changes, particularly among the middle-ranked Qwen and Llama models, and

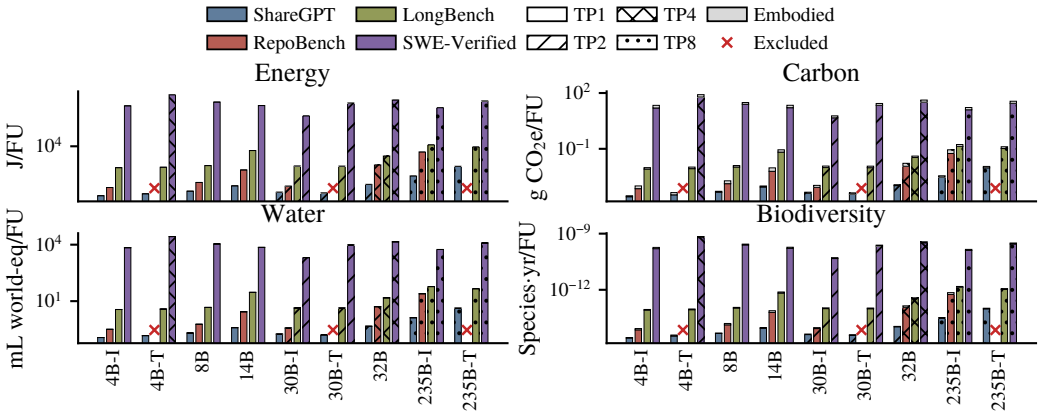


Figure 17: Impact variation across workloads for Qwen models on H100. Each model–workload pair uses its most energy-efficient tensor-parallel configuration. Non-agentic workloads are reported per request, while SWE-bench Verified is reported per successfully completed task. Crosses mark excluded configurations.

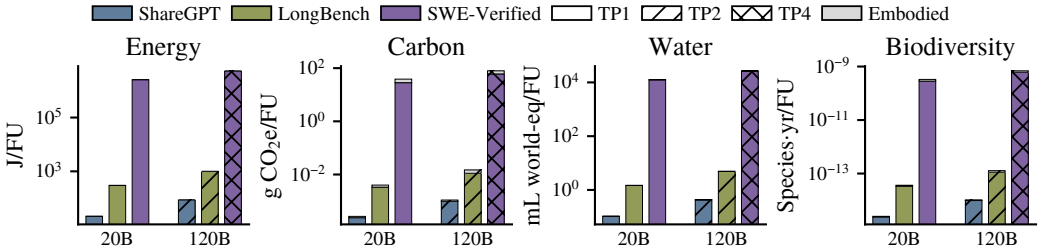


Figure 18: Impact variation across workloads for GPT-OSS models on H100. Each model–workload pair uses its most energy-efficient tensor-parallel configuration. Non-agentic workloads are reported per request, while SWE-bench Verified is reported per successfully completed task.

these changes vary by region. Water largely preserves the configuration ranking induced by energy, while biodiversity introduces only minor additional swaps. Nevertheless, Gemma E4B remains the minimum-impact model across all four dimensions and all three regions, illustrating that substantial ranking disagreement does not necessarily produce optimal-configuration disagreement.

Pairwise ordering flips under controlled hardware and parallelism choices. Disagreement can also arise without changing the model. In Figure 29 ①, we fix Qwen3-4B-Thinking and TP1 and compare GPU platforms. H100 is preferred for energy, water, and biodiversity, whereas A100 is preferred for lifecycle carbon. In Figure 29 ②, we fix Gemma 4 26B on H100 and compare only tensor-parallel settings. TP1 is preferred for energy and water, while TP4 is preferred for lifecycle carbon and biodiversity. These are controlled *pairwise ordering flips*: they show that different dimensions can prefer different hardware or TP choices for the same model, but they do not by themselves imply that either configuration is the global optimum for the full candidate set.

A.5.1 Numerical Validation of the Pairwise Crossover Boundary

Across Figures 5, 27 and 29, the common analytical object is the pairwise crossover condition in Equation (6). For a lower-energy configuration x_1 and a higher-energy configuration x_2, we define the boundary multiple

$$\Delta E_{\mathrm{IT}} = E_{\mathrm{IT}}(x_2, w) - E_{\mathrm{IT}}(x_1, w), \quad \Delta I_{m,\mathrm{emb}} = I_{m,\mathrm{emb}}(x_1, w) - I_{m,\mathrm{emb}}(x_2, w),$$
$$\rho_m = \frac{\Delta I_{m,\mathrm{emb}} / \Delta E_{\mathrm{IT}}}{\alpha_m(r, t)}, \tag{29}$$

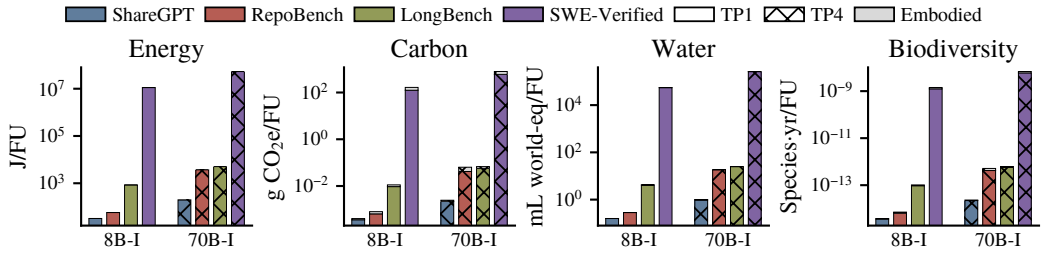


Figure 19: Impact variation across workloads for Llama models on H100. Each model–workload pair uses its most energy-efficient tensor-parallel configuration. Non-agentic workloads are reported per request, while SWE-bench Verified is reported per successfully completed task.

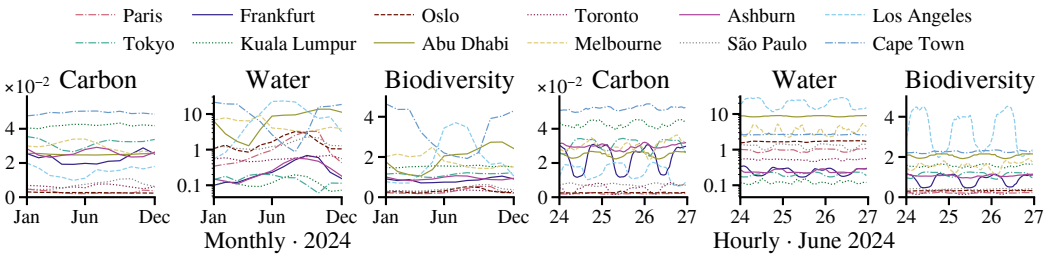


Figure 20: Temporal variation in per-request carbon, water, and biodiversity impact across selected deployment regions under the same serving setting as Figure 4. Left: monthly averages in 2024; right: hourly variation during June 24–26, 2024. Carbon, water, and biodiversity are reported in g CO_2e/request, mL world-eq/request, and 10^{-13} species·yr/request, respectively.

A pairwise ordering flip under dimension m occurs when $\rho_m > 1$, i.e., when the embodied-impact advantage of x_2 exceeds its operational-impact disadvantage under deployment choice (r, t). An optimal-configuration disagreement is a stronger outcome in which such a pairwise crossover changes the minimum-impact configuration among all quality- and SLO-feasible candidates. Table 11 evaluates this condition for representative optimal-configuration disagreement cases and controlled pairwise ordering flips.

The measured preferences agree with the analytical boundary. For the two quality-constrained optimal-choice cases (Figures 5 and 27), lifecycle carbon exceeds the crossover boundary by $1.78\times$ and $1.83\times$, respectively. The controlled comparisons reproduce the dimension-specific disagreements in Figure 29: for ①, only carbon crosses the boundary ($2.93\times$), whereas water and biodiversity remain well below it; for ②, carbon ($5.61\times$) and biodiversity ($2.43\times$) cross, while water does not. The critical-lifetime values provide an equivalent interpretation under our six-year amortization assumption: the corresponding pairwise preference persists while the assumed hardware lifetime remains below the listed threshold.

More generally, these results show why embodied share alone does not determine a decision change: the embodied difference must have the appropriate direction and be sufficiently large relative to both the operational-energy gap and the regional operational intensity. Intuitively, disagreement is easiest to trigger when the deployment choice has a low operational intensity for the dimension under consideration—for example, when carbon intensity is very low for the carbon dimension—because the operational penalty of a higher-energy configuration becomes small enough for embodied-impact differences to overturn the ranking.

A.6 Supplementary Optimization Details

This section provides the additional methodology and sensitivity analysis for the multidimensional routing optimization in the main text. We first describe how the mixed code and conversation workload is constructed from the Azure LLM Inference Dataset and paired with hourly environmental

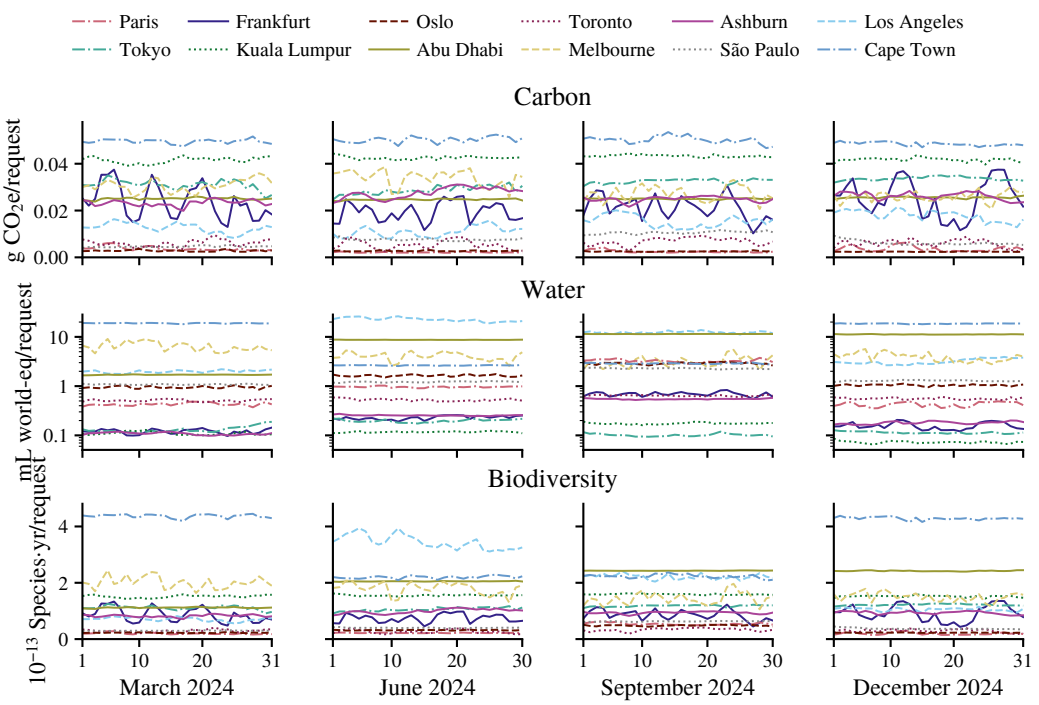


Figure 21: Daily variation in per-request carbon, water, and biodiversity impact across selected deployment regions in March, June, September, and December 2024 under the same serving setting as Figure 4.

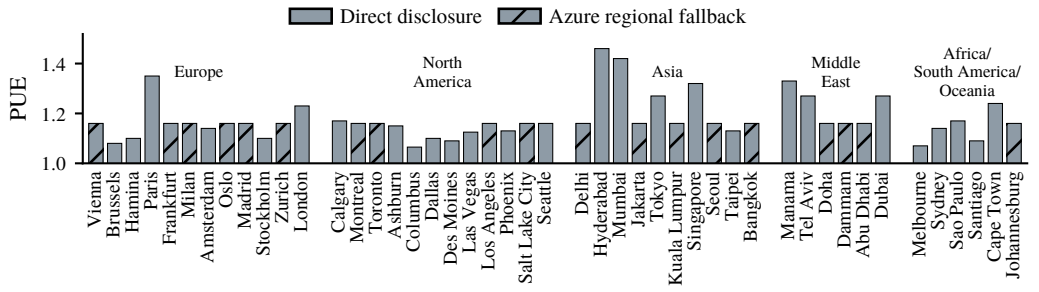


Figure 22: Annual-average PUE values for the modeled regions in Table 9, based on public disclosures from Amazon Web Services (a), Google, and Microsoft Azure.

traces, followed by the synthetic regional-capacity model used to represent heterogeneous accelerator availability. We then give the full routing formulation, including the single-dimension optima and the minimax-regret objective used by PRISM, and report implementation, baseline, solver, and runtime details. Finally, we evaluate sensitivity to workload composition, deployment-region scope, and regional-capacity realizations to test observed reduction in maximum normalized regret persists beyond the main experimental setting.

A.6.1 Trace and Workload Construction

We construct the routing workload from the Azure LLM Inference Dataset 2024. The code stream uses requests from May 10, 2024, and the conversation stream uses requests from May 12, 2024. We retain request arrival time and input/output token lengths, map each request to the corresponding

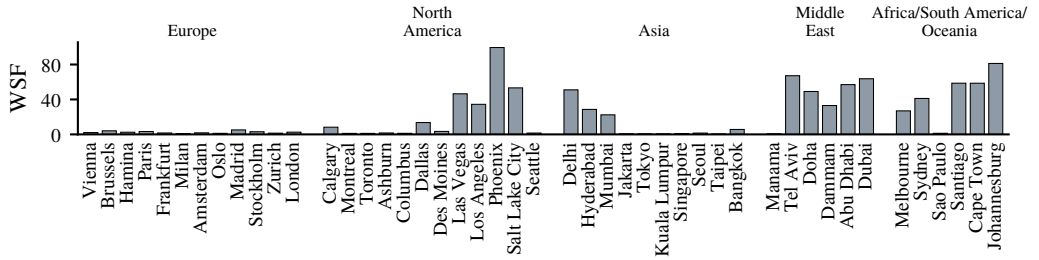


Figure 23: AWARE-2.0 water-scarcity factors for the modeled regions (Seitfudem et al., 2025).

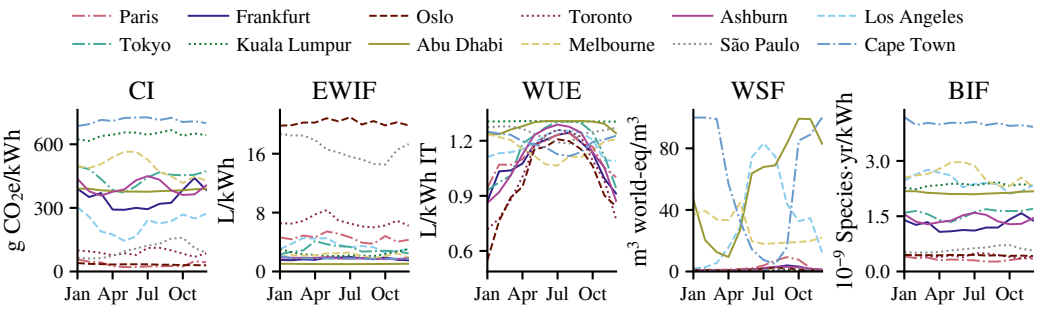


Figure 24: Monthly CI, EWIF, WUE, and BIF for selected modeled regions in 2024, together with the region-specific WSF used for water-scarcity adjustment. CI, EWIF, and BIF follow the observed electricity-grid mix, WUE reflects cooling conditions, while WSF captures regional water scarcity.

measured H100 serving profile, and align the two streams by relative hour. The two streams are scaled to contribute equal offered GPU demand.

The optimization horizon contains 24 one-hour intervals. Requests with the same hour, traffic stream, serving profile, and token-length bins are aggregated into a weighted request group. Each group records the number of equivalent requests and may be divided across regions, corresponding to routing interchangeable requests in different proportions. This produces 1,628 request groups representing approximately 25.2 million request equivalents. Each traffic stream contributes 15.4 million GPU-seconds of offered demand.

We pair the demand trace with hourly environmental factors from June 24, 2024. The trace supplies the within-day demand pattern, while the environmental dataset supplies temporal variation in regional impact intensities. For workload sensitivity, we additionally construct a code-only instance by retaining only the code stream and applying the same preprocessing and capacity-generation procedure.

A.6.2 Regional Capacity Model

We evaluate twelve deployment regions: Abu Dhabi, Melbourne, São Paulo, Toronto, Frankfurt, Paris, Tokyo, Kuala Lumpur, Oslo, Los Angeles, Northern Virginia, and Cape Town, as the regions studied in Figure 20. Because regional accelerator inventories are not publicly available, we synthesize heterogeneous capacity.

For each capacity seed, we draw a lognormal weight for each region with log-space standard deviation $\sigma = 0.5$ and normalize the weights to obtain fixed regional capacity shares. In each hour, total capacity of all regions is provisioned at $1.25\times$ the offered GPU demand. Regional allocations are converted to GPU counts, rounded upward to multiples of 32 H100 GPUs (one rack of 4 DGX H100 systems following NVIDIA's H100 SuperPOD design (NVIDIA, 2023)), and converted back to GPU-seconds. We evaluate seeds 0–19, with all routing policies using identical capacities for a given seed.

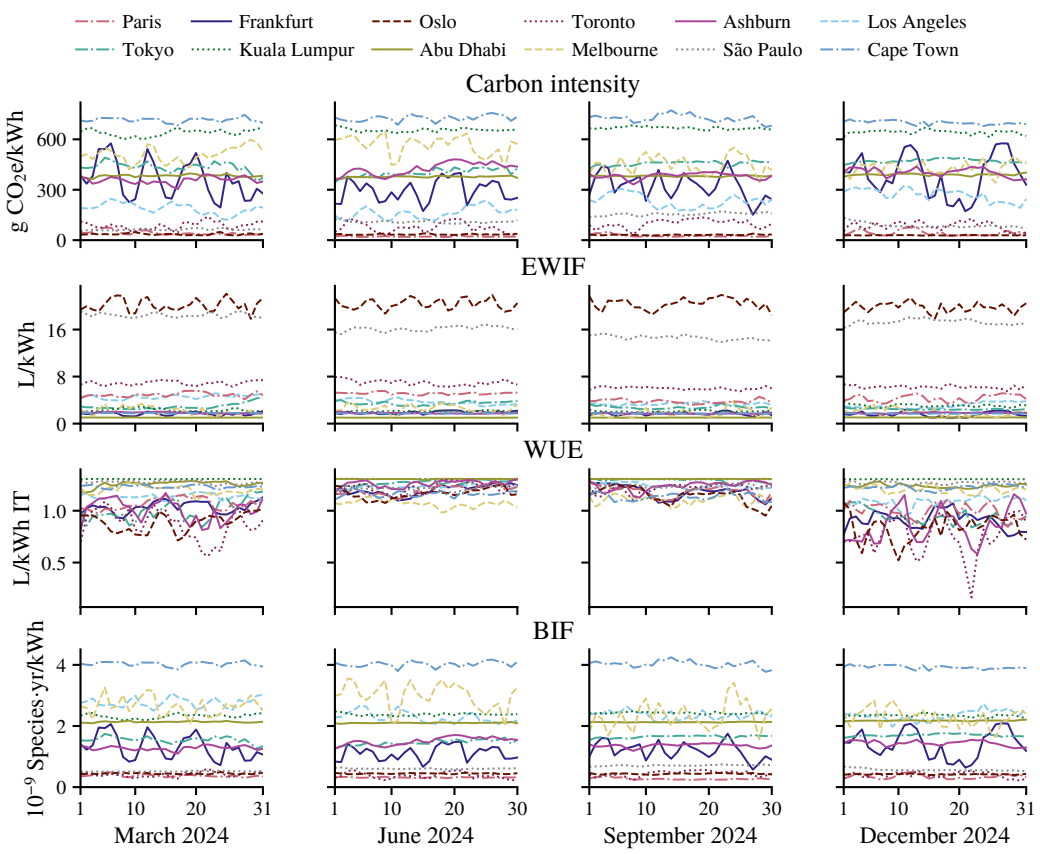


Figure 25: Daily CI, EWIF, WUE, and BIF for selected modeled regions in March, June, September, and December 2024.

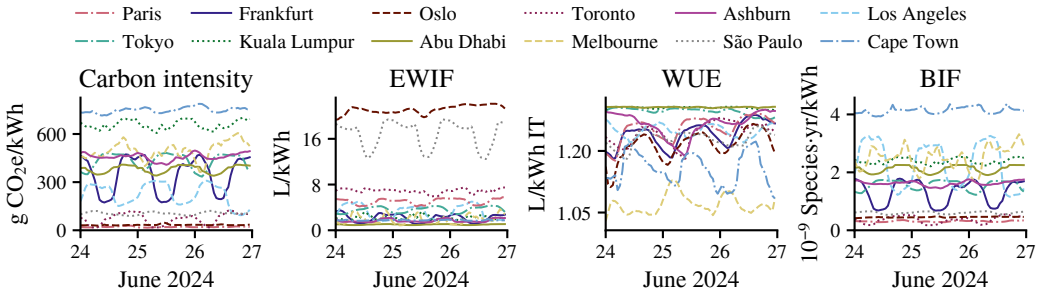


Figure 26: Hourly CI, EWIF, WUE, and BIF for selected modeled regions during June 24–26, 2024.

A.6.3 Routing Optimization

Each workload uses its feasible configuration with the lowest measured IT energy. Let y_{ir} denote the fraction of request group i routed to region r. Each group must be fully assigned:

$$\sum_r y_{ir} = 1, \qquad 0 \le y_{ir} \le 1. \tag{30}$$

Table 10: Ranges of embodied energy contribution to lifecycle energy impact across the evaluated settings in the characterization and disagreement analyses.

Figure	Low (%)	High (%)	Figure	Low (%)	High (%)
Figure 2	1.35	3.72	Figure 14	2.18	3.90
Figure 4	2.23	2.64	Figure 16	1.54	5.30
Figure 5	1.83	2.51	Figure 17	1.36	6.44
Figure 9	1.54	3.83	Figure 18	1.43	4.12
Figure 10	1.48	2.09	Figure 19	1.35	5.89
Figure 11	1.43	2.23	Figure 27	3.83	6.93
Figure 12	1.36	3.72	Figure 28	1.43	4.30
Figure 13	2.84	6.44	Figure 29	1.92	4.30

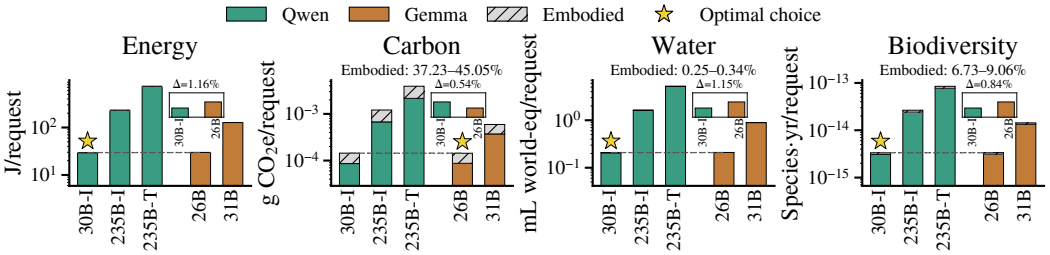


Figure 27: ShareGPT optimal configuration disagreement between dimensions

Let q_i denote the number of equivalent requests in group i, g_i its GPU-seconds per request, and t_i its arrival hour. Regional capacity is constrained by

$$\sum_{i:t_i=t} q_i \, g_i \, y_{ir} \le K_{rt}, \qquad \forall r, t, \tag{31}$$

where K_{rt} is the available GPU-seconds in region r during hour t. Requests are routed within their arrival hour; the experiment does not defer or drop demand. Thus, the time component of each deployment choice is fixed to the request group's arrival hour t_i, and the optimization varies only the deployment region.

For each impact dimension $m \in \{E, C, W, B\}$, we first solve the corresponding single-dimension routing problem to obtain the feasible optimum I_m^*. Let $I_m(y)$ denote the aggregate impact induced by routing assignment y. `PRISM` then solves

$$\begin{aligned} \min_{y,z} \quad & z \\ \text{s.t.} \quad & I_m(y) \le (1+z) I_m^*, \quad \forall m \in \{E, C, W, B\}, \end{aligned} \tag{32}$$

together with the assignment and capacity constraints above. Since aggregated request groups are divisible, the main experiment is a linear program (LP). Representing individual requests with indivisible assignments gives the corresponding mixed-integer formulation (MILP).

All routing policies return assignments to the same impact evaluator, which computes energy, carbon, water, and biodiversity using the accounting model in §2.

A.6.4 Implementation Details

Experimental Configurations. Unless otherwise stated, experiments use a 24-hour horizon with one-hour routing intervals and the twelve-region deployment set. Code and conversation traffic contribute equal offered GPU demand. Total regional capacity in each hour is provisioned at $1.25\times$ offered demand, with regional shares drawn from a lognormal distribution with $\sigma = 0.5$ and held fixed over time. Capacity is rounded upward to 32-H100 units, and results are reported across capacity seeds 0–19. Each workload uses its minimum-IT-energy feasible computing configuration $x_E^*(w)$. All demand must be served in its arrival hour; requests are neither dropped nor deferred.

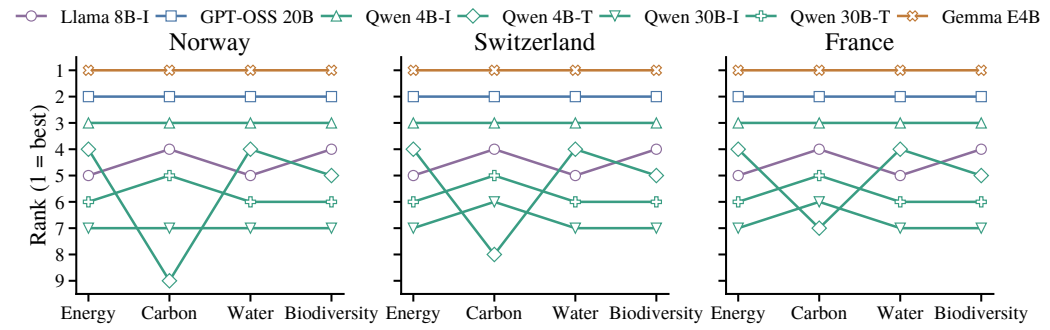


Figure 28: General model-ranking disagreement for ShareGPT across Norway, Switzerland, and France without imposing a quality constraint.

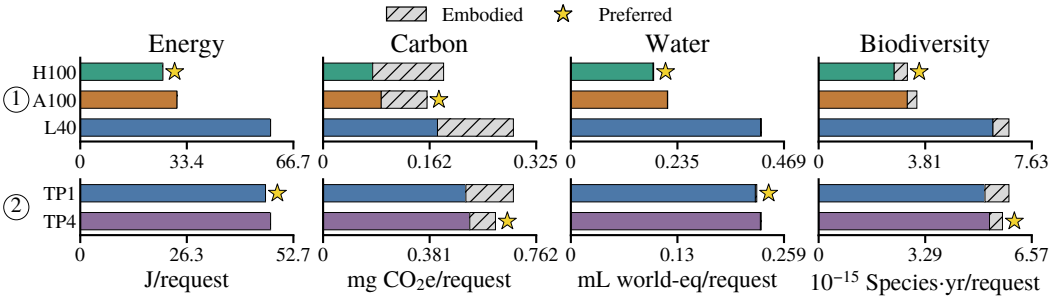


Figure 29: Lifecycle impact disagreement under controlled system choices. ① Qwen3-4B-Thinking on ShareGPT in Norway, comparing H100, A100, and L40 at TP1. ② Gemma 4 26B on ShareGPT in France, comparing H100 TP1 and TP4. Stars mark the preferred choice within each comparison.

Baselines. The single-dimension baselines minimize energy, carbon, water, or biodiversity under the same assignment and capacity constraints. The load-balancing baseline uses regional capacity without environmental information. Our offline WaterWise (Jiang et al., 2025b) adaptation preserves its joint carbon–water objective. Carbon and water are normalized across eligible regions and weighted equally. We evaluate its assignments using the same carbon and stress-adjusted water accounting as the other policies. Since the routing experiment contains neither request deferral nor request-origin information, only the spatial routing component is used.

Solver and Runtime. We implement the optimization in Python 3.12 using SciPy 1.17.1 (Virtanen et al., 2020) with the HiGHS linear-optimization backend (Huangfu & Hall, 2018). The twelve-region instance contains 1,628 request-group assignment constraints, 288 region-hour capacity constraints, and 19,536 routing variables. `PRISM` adds one maximum-regret variable and four regret constraints. On one AMD EPYC 7443 CPU, the complete set of optimization policies for the twelve-region instance requires less than 4 s end-to-end, including constraint construction and impact evaluation. For `PRISM` alone, the optimization solve takes below 0.3 s in our measurements.

A.6.5 Additional Sensitivity Results

We additionally evaluate on the code-only instance defined in §A.6.1. `PRISM` obtains 37.0% worst-case regret, compared with 74.5% for WaterWise, 66.9% for water-only routing, 384.4% for biodiversity-only routing, 466.0% for load balancing, 708.1% for energy-only routing, and 838.6% for carbon-only routing. The result is consistent with the mixed-workload experiment: balancing all four dimensions substantially reduces the maximum normalized regret.

We also examine the influence of regional scope and capacity. We repeat the experiment across 20 capacity seeds under three regional scopes. The *original six-region* set contains Frankfurt, Los Angeles, Tokyo, Kuala Lumpur, Abu Dhabi, and Melbourne, matching the representative locations in Figure 4. The *all-twelve-region* set additionally includes Paris, Toronto, Northern Virginia, Oslo,

Table 11: Numerical validation of the lifecycle crossover boundary. Bold $\rho_m > 1$ indicates a predicted lifecycle ranking reversal.

Workload / configuration pair $(x_1 \to x_2)$	**Dimension**	ΔE_{IT} **(J/req.)**	$\Delta I_{m,\mathrm{emb}}/\Delta E_{\mathrm{IT}}$	$\alpha_m(r,t)$	ρ_m	**Critical lifetime (yr)**
RepoBench (Figure 5) Qwen3 4B-I → Qwen3 30B-I	Carbon	9.422	5.342	2.924	**1.827**	10.96
ShareGPT (Figure 27) Qwen3 30B-I → Gemma 4 26B	Carbon	0.341	5.201	2.924	**1.778**	10.67
Figure 29 ① H100 TP1 → A100 TP1	Carbon	4.482	8.566	2.924	**2.929**	17.58
	Water	4.482	0.106	7.016	0.015	0.09
	Biodiversity	4.482	0.288	1.047	0.275	1.65
Figure 29 ② H100 TP1 → H100 TP4	Carbon	1.248	62.466	11.143	**5.606**	33.64
	Water	1.248	0.622	4.904	0.127	0.76
	Biodiversity	1.248	2.719	1.120	**2.429**	14.57

Units for $\Delta I_{m,\mathrm{emb}}/\Delta E_{\mathrm{IT}}$ and α_m: Carbon: 10^{-6} g CO_2e/J; Water: 10^{-6} L world-eq/J; Biodiversity: 10^{-16} species·yr/J. Critical lifetime is the hardware amortization lifetime at $\rho_m = 1$ with measured throughput fixed.

Table 12: Sensitivity of worst-case regret to geographic scope and capacity seed. Values report median [range] across 20 seeds.

Region set	**WaterWise**	**PRISM**	**Median reduction**
Original 6	43.5% [28.8, 111.2]	37.2% [25.5, 53.5]	19.1%
All 12	157.8% [55.1, 212.3]	75.1% [41.4, 104.0]	50.2%
Concentrated 6	101.7% [57.7, 208.5]	68.1% [48.5, 107.4]	37.9%

Cape Town, and São Paulo and is used for the main optimization result. The *concentrated six-region* set contains Frankfurt, Paris, Los Angeles, Northern Virginia, Toronto, and Tokyo. It provides a less geographically dispersed deployment space concentrated in Europe, North America, and East Asia, allowing us to test whether the result depends on the broader geographic and environmental diversity of the twelve-region set.

Across all three region sets and 60 capacity instances, PRISM achieves lower worst-case regret than WaterWise. The benefit is largest for the twelve-region design space, where greater environmental heterogeneity creates larger trade-offs among the four impact dimensions.

A.7 Extension of Optimization

This section extends the main optimization study beyond its offline, fixed-computing-configuration setting in two directions. First, we evaluate PRISM under rolling-horizon regional routing, where decisions are repeatedly re-optimized using limited and potentially noisy forecasts of future environmental conditions. Second, we consider an agent-heavy workload and jointly optimize computing configuration and deployment region while introducing agent completion time as an additional objective alongside energy, carbon, water, and biodiversity. These extensions test whether the multidimensional optimization remains effective under sequential decision-making and whether its formulation can accommodate configuration choice and performance trade-offs beyond regional routing alone.

A.7.1 Rolling-Horizon Regional Routing

We extend PRISM to rolling-horizon routing using the same 24-hour workload, twelve regions, fixed computing configurations, and seed-0 capacities as the offline experiment. At the beginning of each hour, the scheduler observes the current state and optimizes over a forecast horizon $H \in \{1, 6, 12, 24\}$ hours. It executes only the current-hour allocation and re-optimizes at the next hour; requests remain in their arrival hour.

Table 13: Rolling-horizon PRISM under oracle and noisy environmental forecasts. Gap is the relative increase in maximum regret over full-day offline PRISM.

	Oracle forecast		20% forecast error	
Horizon	**Max. regret**	**Gap**	**Max. regret**	**Gap**
1 h	88.014%	0.697%	88.014%	0.697%
6 h	87.482%	0.088%	87.532%	0.146%
12 h	87.415%	0.012%	87.517%	0.129%
24 h	87.405%	$< 0.001\%$	87.525%	0.138%
Offline PRISM	87.400%	–	–	–
WaterWise-spatial	188.240%	115.36%	–	–

At each optimization step, projected regret combines impacts already realized with forecast impacts over the remaining window and compares them against the corresponding cumulative single-dimension optima. We evaluate both oracle forecasts and noisy environmental forecasts. For the latter, future CI, EWIF, WUE, and BIF values are independently perturbed by mean-one lognormal noise with coefficient of variation 0.20; the current hour is observed exactly. Demand and capacity are assumed known. Full-day offline PRISM provides the reference optimum, and we use the same WaterWise-spatial baseline as in §A.6.4.

As shown in Table 13, even a one-hour oracle horizon remains within 0.70% of the offline optimum in maximum regret. The gap falls to 0.088% at six hours and 0.012% at twelve hours, while the 24-hour horizon reproduces the offline solution to numerical precision. Environmental forecast error has little effect: for horizons containing unobserved future hours, 20% factor noise leaves the gap below 0.15%. In comparison, WaterWise-spatial incurs 188.2% maximum regret. Thus, the multidimensional trade-off obtained offline is largely preserved under sequential routing and remains stable to moderate environmental forecast error.

Across the full 24-hour replay, rolling-horizon optimization requires 0.60–3.21 s of solver time and 12.2–19.5 s end-to-end across the evaluated horizons on the AMD EPYC 7443 system described in §A.6.4.

A.7.2 Joint Optimization with Agentic Workload

We extend the routing experiment to an agent-heavy workload and jointly optimize computing configuration and deployment region. To represent a 2026-style serving mix in which repeated agentic interactions dominate compute demand, the 24-hour workload consists of 60% agentic coding, 30% conversation, 5% long-output summarization, and 5% code completion by offered GPU-seconds. Conversation follows the Azure conversation trace with ShareGPT profiles, summarization uses the LongBench long-output workload, and code completion uses RepoBench. Agentic arrivals follow the hourly shape of the Azure code trace and use SWE-Bench Verified. Defining the mix by GPU demand rather than request count avoids treating a short inference request and a multi-step agentic task as equivalent units of load. We use the same twelve regions, hourly environmental factors, and capacity model as the offline routing experiment.

For agentic coding, the functional unit remains a successfully completed SWE-Bench Verified task. Each arrival can be assigned to any of 16 measured computing configurations, without assuming that the scheduler knows which model will solve an individual task. For configuration x, expected IT energy and GPU demand per successful task are obtained by dividing the full-suite benchmarking totals by its number of successful tasks, while τ_x is the mean completion time among successful tasks. The other workloads retain their fixed measured configurations.

Completion time is introduced as an additional objective because a hard timeout does not distinguish between otherwise feasible agent executions that finish substantially earlier or later. Let y_{ir}^{x} be the fraction of request group i assigned to configuration x and region r, and let ω_i denote its weight. Each request group is fully assigned across configuration–region pairs:

$$\sum_{x\in\mathcal{X}}\sum_{r\in\mathcal{R}} y_{ir}^{(x)} = 1, \qquad y_{ir}^{(x)} \geq 0, \qquad \forall i. \tag{33}$$

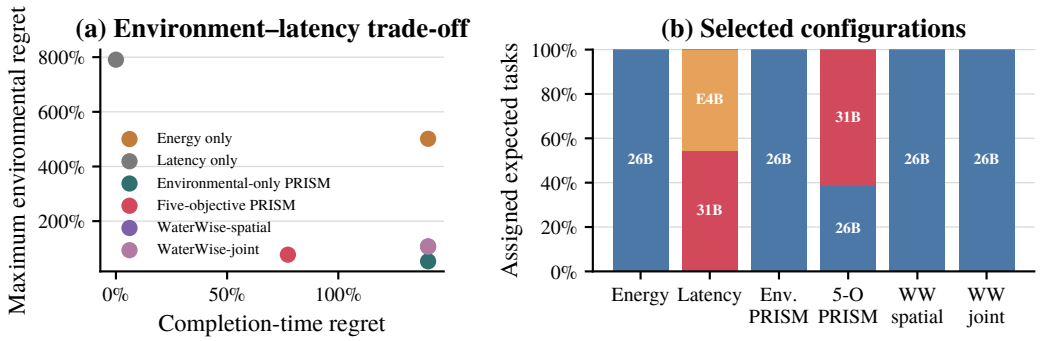


Figure 30: Joint computing-configuration and regional-routing optimization under the agent-heavy workload. (a) Maximum environmental regret versus agentic completion-time regret. (b) Configuration shares selected for the agentic workload.

For the set of agentic groups $\mathcal{A}$, we define mean agentic completion-time objective as

$$L(s) = \frac{1}{\Omega_{\mathcal{A}}} \sum_{i \in \mathcal{A}} \sum_{x,r} \omega_i \, y_{ir}^{(x)} \, \tau_x, \qquad \Omega_{\mathcal{A}} = \sum_{i \in \mathcal{A}} \omega_i. \tag{34}$$

Let L^* be the minimum feasible completion time under the same assignment and capacity constraints. Completion-time regret is

$$R_L(s) = \frac{L(s) - L^*}{L^*}. \tag{35}$$

For each environmental dimension $m \in \{E, C, W, B\}$, we similarly obtain its independent optimum I_m^* and define

$$R_m(s) = \frac{I_m(s) - I_m^*}{I_m^*}. \tag{36}$$

Environmental PRISM minimizes $\max\{R_E, R_C, R_W, R_B\}$, whereas the five-objective formulation jointly minimizes

$$\min_s \max\{R_E(s), R_C(s), R_W(s), R_B(s), R_L(s)\}. \tag{37}$$

We compare these two formulations with energy-only and completion-time-only optimization and with two WaterWise adaptations. WaterWise-spatial first fixes each workload to its minimum expected-energy configuration and optimizes regional carbon and water, while WaterWise-joint allows the same carbon–water objective to choose both configuration and region.

Figure 30 shows that adding completion time changes both the trade-off and the selected configuration. Environmental-only PRISM achieves 53.1% maximum environmental regret but incurs 140.4% completion-time regret. Five-objective PRISM instead limits both to 77.3%. Carbon, stress-adjusted water, and completion-time regrets are binding at the minimax solution, while energy and biodiversity regrets remain lower at 18.0% and 42.1%, respectively. By comparison, energy-only and completion-time-only optimization incur maximum environmental regrets of 501.4% and 790.8%, while WaterWise-spatial and WaterWise-joint obtain 107.4% and 107.2% environmental regret with 140.4% completion-time regret.

The performance objective also changes configuration selection. Environmental PRISM and both WaterWise variants assign all agentic work to Gemma 4 26B. Five-objective PRISM instead assigns 38.4% to Gemma 4 26B and 61.6% to the faster Gemma 4 31B, reducing completion time until its regret reaches the environmental minimax boundary. Completion-time-only optimization shifts further toward faster configurations, assigning 54.4% to Gemma 4 31B and 45.5% to Gemma 4 E4B. Thus, once agent completion time is treated as an objective rather than only a feasibility constraint, computing configuration and regional routing should therefore be optimized jointly.

The resulting continuous linear programs contain 1,247 weighted request groups and 19,284 joint configuration–region assignment variables. Using the same SciPy/HiGHS implementation as §A.6.4, individual solves require 0.05–0.16 s in this experiment.